

2019.09.19

19 September 2019

14:59

今天看看 API 文档 @

学习 java 的第 10 天。

1. java.lang.Object

a) clone()

浅
深

实现 Cloneable 接口, 重写 Object 中的 clone 方法

浅克隆 仅仅复制要克隆的对象, 而不复制它所引用的对象。

深克隆 仅仅复制要克隆的对象, 而不复制它所引用的对象。

深克隆 把要克隆的对象及其所引用的对象。

深克隆 把要克隆的对象及其所引用的对象都复制了一遍。

```
Student s1 = new Student("Jack", 20);
```

```
Student s2 = s1;
```

b) finalize() → FinalizeTest.java

final、finalize、finally?

垃圾回收器准备好释放对象占用的存储空间, 将首先调用其 finalize() 方法, 并且在下一次垃圾回收动作发生时,

才真正回收对象占用的内存。

但是在 Java 如果内存总是充足的，那么垃圾回收可能永远不会进行，也就是说 `finalize()` 可能永远不被执行，显然指望它做收尾工作是靠不住的。

Java 程序有垃圾回收器，所以一般情况下内存问题不用程序员操心。但有一种 JNI (Java Native Interface) 调用 non-Java 程序 (C/C++)，`finalize()` 的工作就是回收这部分的内存。

2、`java.lang.String`：字符串类

常见方法

- ★ `public int length ()`：返回此字符串的长度。
- ★ `public String concat (String str)`：将指定的字符串连接到该字符串的末尾。
- ★ `public char charAt (int index)`：返回指定索引处的 `char` 值，下标从 0 开始。
- ★ `public int indexOf (String str)`：返回指定子字符串第一次出现在该字符串内的索引。
- ★ `public String substring (int beginIndex)`：返回一个子字符串，从 `beginIndex` 开始截取字符串到字符串结尾。
- ★ `public String substring (int beginIndex, int endIndex)`：返回一个子字符串，从 `beginIndex` 到 `endIndex` 截取字符串。含 `beginIndex`，不含 `endIndex`。
- ★ `public boolean contains(CharSequence s)`：判断源字符

串中是否包含s字符串

- ★ `public boolean startsWith(String prefix)`: 判断字符串是不是以某个字符串前缀开始
- ★ `public boolean endsWith(String suffix)`: 判断字符串是不是以某个字符串后缀结尾
- ★ `public char[] toCharArray ()`: 将此字符串转换为新的字符数组。
- ★ `public byte[] getBytes ()`: 使用平台的默认字符集将该 String 编码转换为新的字节数组。
- ★ `public String replace (CharSequence target, CharSequence replacement)`: 将与target匹配的字符串使用replacement字符串替换。
- ★ `public String[] split(String regex)`: 将此字符串按照给定的regex (规则) 拆分为字符串数组。

合

4

c) 正则表达式

匹配正整数: " \\d+"

匹配正小数: " \\d+\\.\\d+"

匹配负整数: " -\\d+"

匹配负小数: " -\\d+\\.\\d+"

匹配保留两位小数的正数: " \\d+\\.\\d{2} "

匹配保留1-3位小数的正数: " \\d+\\.\\d{1,3} "

匹配合法的邮箱: "[a-zA-Z_0-9]+@[a-zA-Z_0-9]+(\\. [a-zA-Z_0-9]+)+"

获取IP地址(192.168.1.100)中的每段数字: "\\."

3. java.util.Date, java.text.SimpleDateFormat 的使用

4. java.util.Calendar

5. System

public static long currentTimeMillis(): 返回以毫秒为单位的当前时间, 与1970年01月01日00:00点之间的毫秒差值。

E O F

张颖捷
2016005851