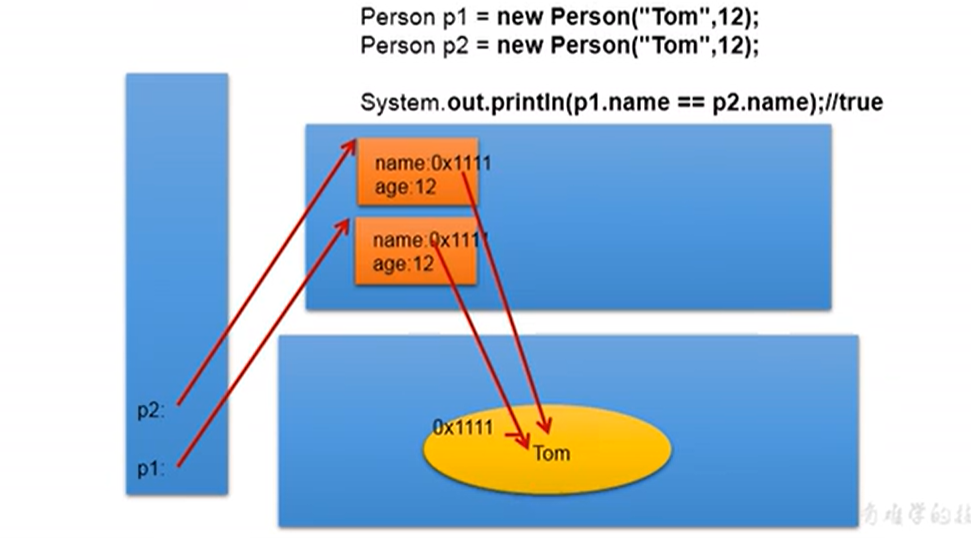
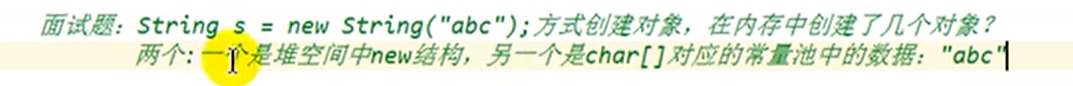
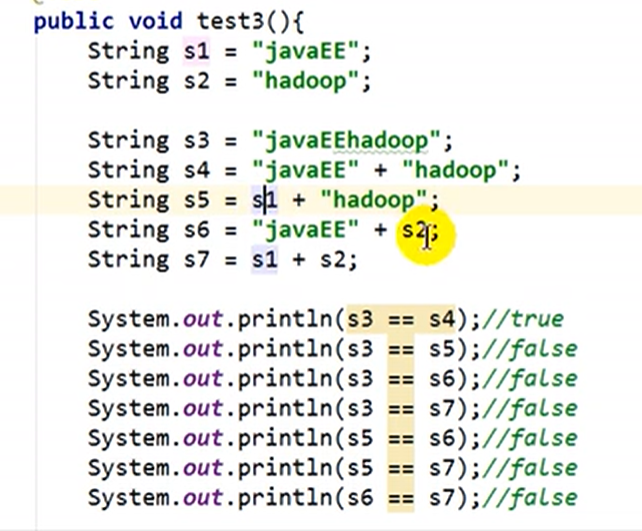
1、

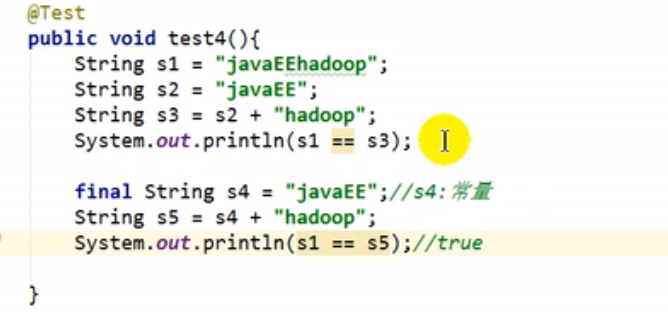
字面量赋值放在方法区的常量池中，唯一性，不可改。多个赋值相同的字符串，变量存放在栈中，实际内容在常量池中。想改变在变量池中重新造一个



New赋值放在堆中。







总结：常量与常量的拼接只发生在常量池中。常量池中只保存一个相同的常量

只要有一个变量参与拼接，那么就存储于堆空间中。

如果调用了方法intern()，那么返回值就在常量池中。

2、String类常用方法

1. int length()
2. char charAt(int index) 返回index处的字符
3. boolean isEmpty()
4. String toLowerCase()转为小写
5. String toUpperCase()大写
6. String trim()返回字符串副本，忽略首位空白
7. Boolean equals(Object obj)
8. Boolean equalsIgnoreCase(Object obj)
9. String concat(String str)拼接
10. int compareTo(String anotherString)比较两个字符串的大小//涉及到字符串排序
11. String substring(int beginIndex)截取
12. String substring(int beginIndex,int endIndex)
13. Boolean endsWith(String suffix)测试此字符串是否以制定的后缀结束
14. Boolean startsWith(String prefix)
15. Boolean startsWith(String prefix，int toffsets)从指定索引开始的字符串是否以制定前缀开始
16. Boolean contains(CharSequens s)当且仅当此字符串包含指定的char值序列时返回true
17. int indexOf(String str)返回指定子字符串在此字符串中第一次出现处的索引
18. int indexOf(String str，int fromIndex)从指定索引位置
19. int lastIndexOf(String str)返回指定子字符串在此字符串中第一次出现处的索引
20. int lastIndexOf(String str，int fromIndex)从指定索引位置

17-20如果未出现均返回-1

1. String replace(char oldChar,char newChar)，全部替换
2. String replace(charSequence target, charSequence replacement)
3. String replaceAll(String regex,String replacement)
4. String replaceFirst(String regex,String replacement)
5. Boolean matches(String regex)
6. String[] split(String regex)拆分字符串
7. String[] split(String regex,int limit)拆分成最多limit个，如果超了，剩下的全部放在最后一个元素中

3、String与基本数据类型转换

Public static int parseInt(String s)

Public String valueOf(int s)

4、String与字符数组转换

1. 字符数组==》字符串 使用字符串的构造器
2. 字符串==》字符数组

Public char[] toCharArray()存放在一个字符数组中

Public void getChars(int srcBegin,int srcEnd,char[] dst,int dstBegin)没整明白

5、String与字节数组转换

1. String==>byte[]

public byte[] getBytes()

Public bytes[] getBytes(tring charsetName)

1. byte[] String

String(byte[])使用平台默认的字符集解码

String(byte[],String CharsetName)使用指定的字符集解码

String(byte[], int offset,int length) 用指定的字节数组一部分， 即从数组起始位置 offset开始取length个字节构造一个字符串对象 。