

JavaApi之String类

1、学习一个新的东西的逻辑发展

- 1)、它是什么?
- 2)、我能用它来做什么?
- 3)、怎么用?
- 4)、为什么这样用?

2、什么是API?

Application Program Interface 应用程序编程接口

API是一组预先定义好的函数（方法）。

java最早是Sun公司开发的。JVM是C语言编写的。现在被Oracle收购。

Oracle（甲骨文）是全球最著名的数据库软件。

java中的API指的是java官方为我们预先编写好的一组类和方法。它包含在我们的java的运行环境中。

java安装后，分为两部分：jdk和jre

jdk: java develop kit java开发工具

jre: java runtime environment java运行环境

这些东西不需要我们自己编写，可以直接使用。

大家学过的：System、Object

例如：

```
System.out.println(123)
Object.clone()
```

3、java中“==”和“equals”

==和equals都用来做比较的。

- 1)、 ==

当比较基本类型的时候，比的是值

当比较对象的时候，比的是地址

- 2)、 equals

equals方法，源自于Object，Object是所有类的父类。

换言之，所有的类，所有的对象，都有equals方法。

大部分情况下，它比较的是值。

只有当重写了equals方法的时候，它会按照重写的逻辑来进行比较。

对于String，Date来说，它们重写了equals方法，比较的是值。

4、值类型的变量和引用类型的变量

什么叫值类型变量？

该变量中存储的就是“值”。比如：8大基本类型

```
int i = 100;
```

什么叫引用类型的变量？

```
public class Student{
    private int stuID;
    private String stuName;

    ...
    ...
}

public static void main(String[] args){
    Student stu = new Student();
}
```

在上述的例子中，stu变量就是一个典型的引用类型的变量。

new Student() 会创建出一个Student类的实例。存在堆中。

声明一个Student类型的变量stu，指向了堆中的Student类的实例。

变量和实例，就相当于：遥控器和电视机。

5、String类

4.1 什么是String？

英文单词本意：串。

java中的String，位于：java.lang.String类。

String表示的是字符串，在java程序中，所有的字符串都是String类的实例。

String的值是不可变的。

java中String是一个非常特殊的对象，从jdk1.7开始，String的对象，也存在与堆中，但是，java对字符串还提供了
一个特殊的内存区域：字符串的常量池。

所有的String的实例，都是指向了该常量池中的值。

4.2 String的常量池

“池”就是一个容器，所谓的常量池，指的就是存放常量的容器。

java在内存中给字符串开辟一块特殊的内存空间，用来保存字符串的常量。

步骤1:

```
String a = "123";
```

在这个例子中，创建了2个。

第一个：变量a，是一个String类的对象。

第二个：常量“123”，java会将“123”这个常量，保存在常量池。

步骤2:

```
String b = "123";
```

当再有一个变量b,也等于“123”的时候，此时：

创建一个变量，b，是String类的一个新的对象

但是，由于b的赋值也是：“123”，

“123”这个字符串在步骤1中，已经保存在常量池中，因此，当执行步骤2的时候，“123”在常量池中已经存在，直接讲String b指向了该常量的地址。

6、String中的构造方法

1、空构造

String ()

使用：String a = new String(); 等价于：String a = "";

2、使用字节数组构造字符串

String(byte[] bytes)

7、String中的常用方法

1、length

String.length() 获取字符串的长度。

2、charAt

返回指定位置的字符

```
System.out.println(a.charAt(0));
```

下标如果小于0，或者大于等于length，报错。

3、indexOf

返回指定子串在该字符串中第一次出现的位置

如果找到，返回第一次出现的下标

找不到，返回-1

4、字符串拼接

4.1 "+"

```
System.out.println("abc" + 53.3);
```

4.2 concat方法

```
System.out.println("abc".concat("xyz"));
```

4.3 format

引子：

打印一个学生对象的值：

```
String name = "麻子";
```

```
int age = 16;
```

```
String sex = "女";
```

```
System.out.println("姓名 = " + name + ", 年龄 = " + age + ", 性别 = " + sex);
```

弊端：写起来非常的麻烦，它采用的是字符串的拼接的方式来做。

可以考虑使用String.format来改造

```
String name = "麻子";
int age = 16;
String sex = "女";
boolean x = false;

System.out.println(String.format("姓名=%s,年龄=%d,性别=%s,布尔=%b", name, age, sex, x));
```

5、字符串的转换

5.1 大小写转换

toLowerCase 转小写
toUpperCase 转大写

5.2 valueOf

String.valueOf (参数)
将其他类型的数据, 转换成String类型。

```
char[] data = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'};
System.out.println(String.valueOf(data));
System.out.println(String.valueOf(data, 6, 2));
```

6、字符串的截取

6.1 substring

```
String str = "abcdefg";
System.out.println(str.substring(3)); //从下标为3的字符开始, 取到最后。包括3
System.out.println(str.substring(3, 6)); // 从下标为3开始, 截取到下标6, 包括3, 不包括6.
```

6.2 split

字符串的分割

```
String str =
"C:\\ProgramData\\Oracle\\Java\\javapath;%SystemRoot%\\system32;%SystemRoot%;%SystemRoot%\\System32\\Wbem;%SYSTEMROOT%\\System32\\WindowsPowerShell\\v1.0\\;C:\\Program
Files\\Java\\jdk1.8.0_77\\bin";
String[] split = str.split(";");
System.out.println(split);

String a = "abc012304320213";
String[] split1 = a.split("0");
for (String string : split1) {
    System.out.println(string);
}
```

6.3 startswith 和 endswith

判断字符串是否以指定的字符串开始，或者结束。

```
String a = "abc012304320213";

System.out.println(a.startsWith("abc"));
System.out.println(a.endsWith("3"));
```

6.4 replace

字符串的替换

6.5 字符串的比较

compareTo

```
String a = "abcda";
String b = "abcd";

System.out.println(a.compareTo(b));
```

比较的逻辑：

- 1、按照字典顺序比较两个字符串。
什么叫字典顺序？两个字符串——对应的比较。按照下标。
- 2、比较的逻辑
先取两个字符串的第一个字符，进行比较，
如果不相同，返回两个字符的Ascii码的差值（前面的字符-后面的字符得到的差值）。
后面的就不再进行比较。
如果相同，则取双方的第二个字符进行比较，逻辑同上。依次类推直到比较出两个不同的字符。

特殊情况:

- 1、两个字符串完全一致, 返回0
- 2、如果A字符串的长度小于B字符串的长度, 并且A有的, 与B完全一致, 返回-1
表示A<B
- 3、如果A字符串的长度大于B字符串的长度, 并且B有的, 与A完全一致, 返回1
表示A>B

总结起来一句话: `a.compareTo(b)`, 如果0, 则`a=b`, 如果为负数, `a<b`, 如果是正数, `a>b`

7、StringBuffer

回顾: String是一个不可变的字符串。

StringBuffer是一个可变的字符串缓冲区, 它提供了字符串的追加, 替换, 删除, 插入等操作。

7.1 append

追加字符串

7.2 delete和deleteCharAt

deleteCharAt删除指定位置的字符

delete删除制定长度的字符串, 包括起点, 不包括终点。

7.3 insert

在字符串的指定位置, 插入制定的数据

7.4 reverse

反转字符串

7.5 总结

1、String是一个不可变的, StringBuffer是可变的

2、当大量的字符串拼接的时候, StringBuffer的性能远远的好于String。

因此, 当我们需要改变字符串的内容的时候, 第一个想到的就应该使用StringBuffer

3、当做为参数传递的时候, String是值传递, 在方法内部修改参数的值, 外部是不会收到影响的。

stringBuffer做为参数传递, 是引用传递, 方法内部修改了参数的值, 外部会收到影响。

8、查看字节码文件

`javap -verbose` class文件的路径

例如：

```
javap -verbose E:/workspace/demo/Demo.class
```

9、StringBuilder

与StringBuffer使用上基本一致。不同的是：StringBuilder是线程不安全，StringBuffer是线程安全的。

既然StringBuffer是保证了线程的安全，意味这StringBuffer的性能比StringBuilder低。