Lab2 字符串、数组和文件I/O

- 字符串
- 数组
- 文件I/O

吴昊伦

51265901074@stu.ecnu.edu.cn

创建字符串

1. 先声明后赋值

```
String s0;
s0 = "hello";
```

2. 当然也可以在同一行代码内完成

```
String s0 = "hello";
```

3. 另外还可以使用new关键字来创建String

```
String s0 = new String("hello");
```

4. 甚至这样创建String也是可以的

```
char[] c1 = {'h','e','l','l','o'};
String s0 = new String(c1);
```

字符串

字符串的常用操作

1. 获取字符串长度

```
String s0 = "hello";
System.out.println(s0.length());  // 5
```

```
String s0 = "你好";
System.out.println(s0.length());  // 2
```

字符串

字符串的常用操作

2. 拼接字符串

尝试在主函数执行以下代码

```
String s0 = "hello";
String s1 = ", world";
System.out.println(s0.concat(s1));

System.out.println(10 + 'a');
System.out.println(10 + "a");

Sytem.out.println(s0.concat(10));
```

区别：`concat()`的两侧都必须是`String`，不能是`null`或其他类型

字符串

字符串的操作

3. `equals()`：字符串比较

```
String s0 = "hello";
String s1 = "hello";
String s2 = "world";
System.out.println(s0.equals(s1));  // true
System.out.println(s0.equals(s2));  // false
```

`equals()`用于比较字符串的内容，而`==`用于比较引用：

```
System.out.println(s0 == s1);  // true
System.out.println(s1 == s2) // false
```

字符串

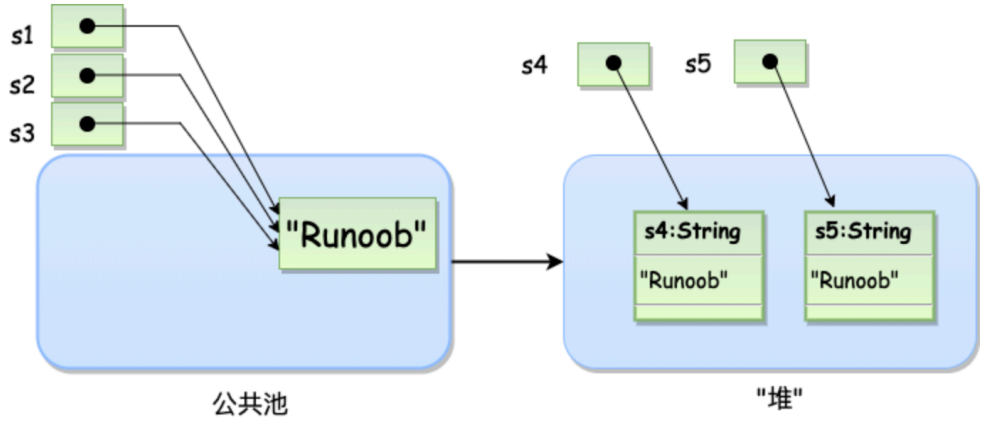
new String和String的区别?

```
String s0 = "hello";
String s1 = "hello";
String s2 = new String("hello");
String s3 = new String("hello");
System.out.println(s0 == s1);
System.out.println(s2 == s3);
System.out.println(s0 == s2);
```

Java中使用String直值创建的字符串存储在常量池中，而使用new关键字创建的String对象则存储在堆中。对于内容相同的字符串，在常量池中永远只有一份，在堆中有多份。

字符串

new String和String的区别?



尝试执行以下代码，思考`s1 = s1.intern();`的作用

```
String s1 = "Runoob";
String s4 = new String("Runoob");

System.out.println(s1==s4);
s4 = s4.intern();
System.out.println(s1==s4);
```

数组

一维数组

声明数组

```
int[] arr1;  // 推荐
int arr2[];
```

创建数组

```
int[] arr1 = new int[10]; //默认值是0
int[] arr2 = new int[] {1,2,3,4,5};
int[] arr3 = {1, 2, 3, 4, 5};
```

与C语言类似的是，数组被创建后，其长度不能被修改

与C语言不同的是，数组初始化存在默认值（例如上面的例子中，`arr1`的每个元素默认都是0）

数组

二维数组

声明数组

```
int[][] arr1;  // 推荐
int arr2[][];
```

初始化

```
int[][] arr1 = {{1,2,3},{4,5,6}};
int[][] arr2 = new int[3][];
int[][] arr3 = new int[3][3];
```

点击这里做一个小测试吧!

数组

Java中的动态数组-ArrayList

1. 初始化

```
ArrayList<E> arr = new ArrayList<>();
```

2. 添加项

```
arr.add("hello");
arr.add("world");
```

数组

Java中的动态数组-ArrayList

3. 根据下标得到item

```
arr.get(0); // hello
```

4. 删除某项

```
arr.remove("hello");
```

5. 数组长度

```
System.out.println(arr.size()); // 1
```

数组

ArrayList

```
# python中的list
arr = []
arr.append("hello")
arr.append("world")
arr.append("!")

for str in arr:
    print(str)


arr.remove("second")
print(arr)
```

文件I/O

流

- 流（Stream）：一组有顺序的、有起点和终点的字节集合，是对数据传输的总称。数据在两个对象之间的传输就称为流。可以将流理解为连接两个水桶的水管。
- 按照流的传输方法，可以分为**输入流**和**输出流**, 按照流的处理对象，又可以分为**字节流**和**字符流**。
- 流的基本操作有读操作和写操作，从流中取得数据的操作称为读操作，向流中添加数据的操作称为写操作。对输入流只能进行读操作，对输出流只能进行写操作
- Java中与IO流相关的共有40多个类...hhh

文件I/O - FileIO.java

方法	说明
`void writeStringToFile(String str, String fileName)`	将一个字符串追加地写入指定文件
`char getCharFromFile(int pos, String fileName)`	将读取指定文件的第pos个字符
`String getLineFromFile(int pos, String fileName)`	读取文件的第pos行的字符串
`String[] getAllLinesFromFile(String fileName)`	读取文件中所有行中的字符串,并按照顺序储存在String类型数组中返回

附录 - 字符串常用方法

方法	描述
`int length()`	返回字符串的长度
`String toUpperCase()`	将串中字符变成大写
`String toLowerCase()`	将串中字符变成小写
`char charAt(int index)`	返回位置index处的字符
`String substring(int s, int e)`	返回从位置s到e的字符子串[s,e)
`String substring(int s)`	返回从位置s到末尾的字符子串

附录 - 字符串常用方法

方法	描述
`int indexOf(String s)`	返回首次出现字符串s的位置
`int indexOf(String s,int i)`	返回在位置i之后首次出现s的位置
`String trim()`	返回一个新串，去除前后空白字符
`String replace(String a, String b)`	返回一个新串，将a替换为b
`String trim()`	返回一个新串，去除前后空白字符
`String replace(String a, String b)`	返回一个新串，将a替换为b