

| 今天学习目标

- * 能够使用String对象存储字符串

- * String str="";

- * String str1=new String("");

- * char[] chs={};

- String str2=new String(chs);

- * 能够查看String源码理解String本质是char数组

- private final char value[];

- * 能够查看String源码理解String不可变性

- private final char value[];

- * 能够掌握String常见的方法

- * charAt , compareTo , compareToIgnoreCase, concat, contains

- * startsWith, equals(==), indexOf, lastIndexOf,

- length, split

- * toCharArray, toLowerCase, trim, valueOf,

- format,substring, replace

- * 能够通过API的文档查看String的相关的方法

- * 能够实现注册新用户，要求密码长度不能小于6位

- * 能够实现注册成功后，实现登录验证。用户名为“xiaohai”，密码为“123456”

- * 能够实现判断判断邮箱格式是否正确

- * 能够实现判断.java文件名是否正确

- * 能够理解字符串常量池

-
- * 回顾

- * 面向对象

- * 面向对象是软件开发过程的方法论。

- * 语法，思维

- * 面向对象与面向过程的区别

- * 财务报销和审核

- * 类的编写者和类的使用者

- * 类和对象

- * 类：抽象的概念

- * 属性+方法

- * 对象：具体的概念

- * 方法：无参方法，有参方法

- * 包：import，ArrayList，LinkedList，String (java.lang.*)

- * ctrl+shift+o

- * 注释

- * // 单行注释 /* */ 多行注释 /** */ 文档注释 (javadoc)

- * 堆栈图

- * 基本数据类型作为方法参数

- * 交换两个数

- * 引用类型作为方法参数

- * 交换两个数

- * 能够使用String对象存储字符串

- * String str="HelloWorld";

- * String s1=new String("HelloWorld");

- * char[] chs= {'a','b','c'};

- String str2=new String(chs);

- * 能够查看String源码理解String本质是char数组

- * String本质char数组+对char数组操作

- * 能够查看String源码理解String不可变性

* `private final char value[];`(常量)

* 意味着它是不可改变的

* 能够掌握String常见的方法

* `charAt`, `compareTo`, `compareToIgnoreCase`, `concat`, `contains`

* `startsWith`, `equals`, `indexOf`, `lastIndexOf`, `length`, `split`

* `toCharArray`, `toLowerCase`, `trim`, `valueOf`

* `public char charAt(int index)`

* 返回指定索引处的 char 值

* "Hello"--->e-->

```
16
17     System.out.println(str);
18     System.out.println(str1);
19     System.out.println(str2);
20
21     char c=str2.charAt(1);
22     System.out.println(c);
23 }
24
```

String本质就是char数组

<terminated> Test3 (8) [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (2018年8月8日 上午11:5

Hello
Hello
Hello
e

源码的角度:

```
public char charAt(int index) {
    if ((index < 0) || (index >= value.length)) {
        throw new StringIndexOutOfBoundsException(index);
    }
    return value[index];
}
```

char数组

* `public int compareTo(String anotherString)`

* 按字典顺序比较两个字符串

* "sun", "music", "moon"

* 返回值的解释

* 返回值等于零：两个字符串相等

* 返回值小于零：字符字典顺序比参数字符串前

* 返回值大于零：字符字典顺序比参数字符串后

```
public int compareTo(String anotherString) {  
    int len1 = value.length;  
    int len2 = anotherString.value.length;  
    int lim = Math.min(len1, len2);  
    char v1[] = value;  
    char v2[] = anotherString.value;  
  
    int k = 0;  
    while (k < lim) {  
        char c1 = v1[k];  
        char c2 = v2[k];  
        if (c1 != c2) {  
            return c1 - c2;  
        }  
        k++;  
    }  
    return len1 - len2;  
}
```

* `public int compareToIgnoreCase(String str)`

* 按字典顺序比较两个字符串，不考虑大小写

* `public String concat(String str)`

* 类似于加号

* 将指定字符串连接到此字符串的结尾

```

public String concat(String str) {
    int otherLen = str.length();
    if (otherLen == 0) {
        return this;
    }
    int len = value.length;
    char buf[] = Arrays.copyOf(value, len + otherLen);
    str.getChars(buf, len);
    return new String(buf, true);
}

```

* `public boolean contains(CharSequence s)`

* 我来中国---》中国是否包含在这句话里

```

String s6="我爱中国";
// 判断中国是否在s6
if(s6.contains("中国d")) {
    System.out.println("包含中国");
}else {
    System.out.println("不包含中国");
}

```

* `public boolean startsWith(String prefix)`

* 测试此字符串是否以指定的前缀开始。

* `public boolean endsWith(String suffix)`

* 测试此字符串是否以指定的后缀结束。

```

String s7="LoginMain";
String s8="HelloWorld.java";
System.out.println(s7.startsWith("Login"));
System.out.println(s8.endsWith("java"));

```

* `public boolean equals(Object anObject)`

* 将此字符串与指定的对象比较。（内容的比较）

* 源码分析

```
public boolean equals(Object anObject) {  
    if (this == anObject) {  
        return true;  
    }  
    if (anObject instanceof String) {  
        String anotherString = (String)anObject;  
        int n = value.length;  
        if (n == anotherString.value.length) {  
            char v1[] = value;  
            char v2[] = anotherString.value;  
            int i = 0;  
            while (n-- != 0) {  
                if (v1[i] != v2[i])  
                    return false;  
                i++;  
            }  
            return true;  
        }  
    }  
    return false;  
}
```

对比两个字符数组，只要字符数组里面有个字符不相等，就返回false

* "==" 比较是字符串的地址

* `public boolean equalsIgnoreCase(String anotherString)`

* 将此 String 与另一个 String 比较，不考虑大小写。

```
public boolean equalsIgnoreCase(String anotherString) {  
    return (this == anotherString) ? true  
        : (anotherString != null)  
        && (anotherString.value.length == value.length)  
        && regionMatches(true, 0, anotherString, 0, value.length);  
}
```

长度相等

字符匹配

```

while (len-- > 0) {
    char c1 = ta[to++];
    char c2 = pa[po++];
    if (c1 == c2) {
        continue;
    }
    if (ignoreCase) {
        // If characters don't match but case may be ignored,
        // try converting both characters to uppercase.
        // If the results match, then the comparison scan should
        // continue.
        char u1 = Character.toUpperCase(c1);
        char u2 = Character.toUpperCase(c2);
        if (u1 == u2) {
            continue;
        }
    }
}

```

先转换成大写字符比较

```

if (ignoreCase) {
    // If characters don't match but case may be ignored,
    // try converting both characters to uppercase.
    // If the results match, then the comparison scan should
    // continue.
    char u1 = Character.toUpperCase(c1);
    char u2 = Character.toUpperCase(c2);
    if (u1 == u2) {
        continue;
    }
    // Unfortunately, conversion to uppercase does not work properly
    // for the Georgian alphabet, which has strange rules about case
    // conversion. So we need to make one last check before
    // exiting.
    if (Character.toLowerCase(u1) == Character.toLowerCase(u2)) {
        continue;
    }
}
return false;

```

转换成小写字符比较

* `public static String format(String format, Object... args)`

* 格式化字符串

* 参考：<https://www.cnblogs.com/happyday56/p/3996498.html>

* https://blog.csdn.net/qq_25925973/article/details/54407994

```

4 public static void main(String[] args) {
5     String str=null;
6     str=String.format("Hi,%s", "王力");
7     System.out.println(str);
8     str=String.format("Hi,%s:%s.%s", "王南","王力","王张");
9     System.out.println(str);
10 }
11 }
12

```

Problems @ Javadoc Declaration Search Console

<terminated> Test4 (8) [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (2018年8月8日 下午2:55:38)

Hi,王力

Hi,王南:王力.王张



```

1 public static void main(String[] args) {
2     String str=null;
3     str=String.format("Hi,%s", "王力");
4     System.out.println(str);
5     str=String.format("Hi,%s:%s.%s", "王南","王力","王张");
6     System.out.println(str);
7 }

```

* `public 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。。


```

4 public static void main(String[] args) {
5     String str="abcdce";
6     int index = str.indexOf("c");
7     System.out.println(index);
8 }
9 }
10

```

第一次出现的位置

Problems @ Javadoc Declaration Search Console

<terminated> Test5 (7) [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (2018年8月8日 下午3:02:37)

2

* `public int lastIndexOf(String str)`

* 返回指定子字符串在此字符串中最右边出现处的索引

```

5 public class Test5 {
6     public static void main(String[] args) {
7         String str="abcdce";
8         int index = str.indexOf("c");
9         int lastIndex=str.lastIndexOf("c");
10        System.out.println(index);
11        System.out.println(lastIndex);
12    }
13 }

```

最后出现的位置

Problems @ Javadoc Declaration Search Console

<terminated> Test5 (7) [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (2018年8月8日 下午3:04:52)

2

4

* `public int length()`

* 返回此字符串的长度

```

public int length() {
    return value.length;
}

```

字符数组的长度

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

* 返回一个新的字符串，它是通过用 newChar 替换此字符串中出现的所有 oldChar 得到的

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

* 使用指定的字面值替换序列替换此字符串所有匹配字面值目标序列的子字符串

```
4     public static void main(String[] args) {  
5         String str="大家坏才是真的坏";  
6         // 把"坏" 替换成 "好"  
7         String newStr = str.replace('坏', '好');  
8         System.out.println(newStr);  
9     }  
10 }  
11
```

Problems @ Javadoc Declaration Search Console

<terminated> Test5 (7) [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (2018年8月8日 下午5
大家好才是真的好

* `public String[] split(String regex)`

* 根据给定正则表达式的匹配拆分此字符串

```

4 public static void main(String[] args) {
5     String str="apple,banana,pear,watermelon";
6     // 对进行拆分成字符串数组
7     String[] fruits = str.split(",");
8     for(String fruit:fruits) {
9         System.out.println(fruit);
10    }
11 }
12 }
13

```

Problems @ Javadoc Declaration Search Console

<terminated> Test7 (7) [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (2018年8月8日 下午5:25:30)

```

apple
banana
pear
watermelon

```

* `public String substring(int beginIndex, int endIndex)`

* `public String substring(int beginIndex)`

* 返回一个新字符串，它是此字符串的一个子字符串。

```

3 public class Test8 {
4     public static void main(String[] args) {
5         String str="www.baidu.com";
6         String substring = str.substring(0, 3); // [0,3)
7         System.out.println(substring);
8         String str2=str.substring(3); // 从下标为3的开始取
9         System.out.println(str2);
10    }
11 }

```

从下标为0的开始截取，截取到下标2的位置，不包含下标3

从下标为3开始包含下标3，截取之后的字符

Problems @ Javadoc Declaration Search Console

<terminated> Test8 (6) [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (2018年8月8日 下午5:29:54)

```

www
.baidu.com

```

* `public char[] toCharArray()`

* 将此字符串转换为一个新的字符数组

* `public String toLowerCase()`

* 转换为小写

- * `public String toUpperCase()`
 - * 转换为大写
- * `public String trim()`
 - * 返回字符串的副本，忽略前导空白和尾部空白
- * `public static String valueOf()`
 - * 把其他类型转换成字符串类型
- * 能够通过API的文档查看String的相关的方法
- * 能够实现注册新用户，要求密码长度不能小于6位
 - * 1 注册新用户，要求密码长度不能小于6位

```
请输入用户名: xiaohei
请输入密码: 123
密码长度不能小于6位!
```

```
public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    String name;
    String pwd;
    System.out.print("请输入用户名: ");
    name = input.next();
    System.out.print("请输入密码: ");
    pwd = input.next();
    if (pwd.length() >= 6) {
        System.out.print("注册成功! ");
    } else {
        System.out.print("密码长度不能小于6位! ");
    }
}
```

```
1 public static void main(String[] args) {
2     Scanner input = new Scanner(System.in);
3     String name;
4     String pwd;
5     System.out.print("请输入用户名: ");
6     name = input.next();
7     System.out.print("请输入密码: ");
8     pwd = input.next();
9     if (pwd.length() >= 6) {
10         System.out.print("注册成功! ");
11     } else {
12         System.out.print("密码长度不能小于6位!");
13     }
14 }
```

* 能够实现注册成功后，实现登录验证。用户名为“xiaohei”，密码为“123456”

* 注册成功后，实现登录验证。用户名为“xiaohei”，密码为“123456”

请输入用户名: xiaohei
请输入密码: 123456
登录成功!

请输入用户名: xiaohei
请输入密码: 123
用户名或密码不匹配，登录失败!

```

public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    String name,pwd;
    System.out.print("请输入用户名: ");
    name=input.next();
    System.out.print("请输入密码: ");
    pwd=input.next();
    if("xiaohei".equals(name) && "123456".equals(pwd) ){
        System.out.print("登录成功! ");
    }else{
        System.out.print("用户名或密码不匹配, 登录失败! ");
    }
}

```

```

1 public static void main(String[] args) {
2     Scanner input = new Scanner(System.in);
3     String name,pwd;
4     System.out.print("请输入用户名: ");
5     name=input.next();
6     System.out.print("请输入密码: ");
7     pwd=input.next();
8     if("xiaohei".equals(name) && "123456".equals(pwd) ){
9         System.out.print("登录成功! ");
10    }else{
11        System.out.print("用户名或密码不匹配, 登录失败! ");
12    }
13 }

```

* 细节：“==”和equals()有什么区别呢？

* equals():检查组成字符串内容的字符是否完全一致

* ==:判断两个字符串在内存中的地址，即判断是否是同一个字符串对象

* 能够实现判断邮箱格式是否正确

* 88888123@qq.com (正则表达式)

* 两个条件：

* 是否包含@

* 是否“.”在@后面

请输入邮箱: 386777911@qq.com
邮箱格式正确

```
6 public static void main(String[] args) {
7     Scanner input=new Scanner(System.in);
8     System.out.print("请输入邮箱: ");
9     String email=input.next();
10    // 两个条件:
11    // * 是否包含@
12    // * 是否“.”在@后面
13    // 88888123@qq.com
14    if(email.indexOf("@")!=-1 && email.indexOf(".")>email.indexOf("@")) {
15        System.out.println("邮箱格式正确");
16    }else {
17        System.out.println("邮箱格式错误");
18    }
19 }
20 }
21
```

Problems Javadoc Declaration Search Console

<terminated> Test9 (6) [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (2018年8月8日 下午7:16:28)

请输入邮箱: 386777911@qq.com
邮箱格式正确

```
1 public static void main(String[] args) {
2     Scanner input=new Scanner(System.in);
3     System.out.print("请输入邮箱: ");
4     String email=input.next();
5     // 两个条件:
6     // * 是否包含@
7     // * 是否“.”在@后面
8     // 88888123@qq.com
9     if(email.indexOf("@")!=-1 && email.indexOf(".")>email.indexOf("@")) {
10        System.out.println("邮箱格式正确");
11    }else {
12        System.out.println("邮箱格式错误");
13    }
14 }
```

- * 能够实现判断.java文件名是否正确
 - * 判断.java文件名是否正确
 - * 最后"."是否存在
 - * "."不能是第一个
 - * 判断"."之后的字符串是不是"java"

请文件名: HelloWorld.java
文件合法

```

public static void main(String[] args) {
    Scanner input=new Scanner(System.in);
    System.out.print("请文件名: ");
    String fileName=input.next();
    // * 最后"."是否存在
    // * "."不能是第一个
    // * 判断"."之后的字符串是不是"java"
    int index=fileName.lastIndexOf(".");
    String suffix=fileName.substring(index+1, fileName.length());
    if(index!=-1 && index!=0 && "java".equals(suffix)) {
        System.out.println("文件合法");
    }else {
        System.out.println("文件不合法");
    }
}

```

```

1 public static void main(String[] args) {
2     Scanner input=new Scanner(System.in);
3     System.out.print("请文件名: ");
4     String fileName=input.next();
5     // * 最后"."是否存在

```



```

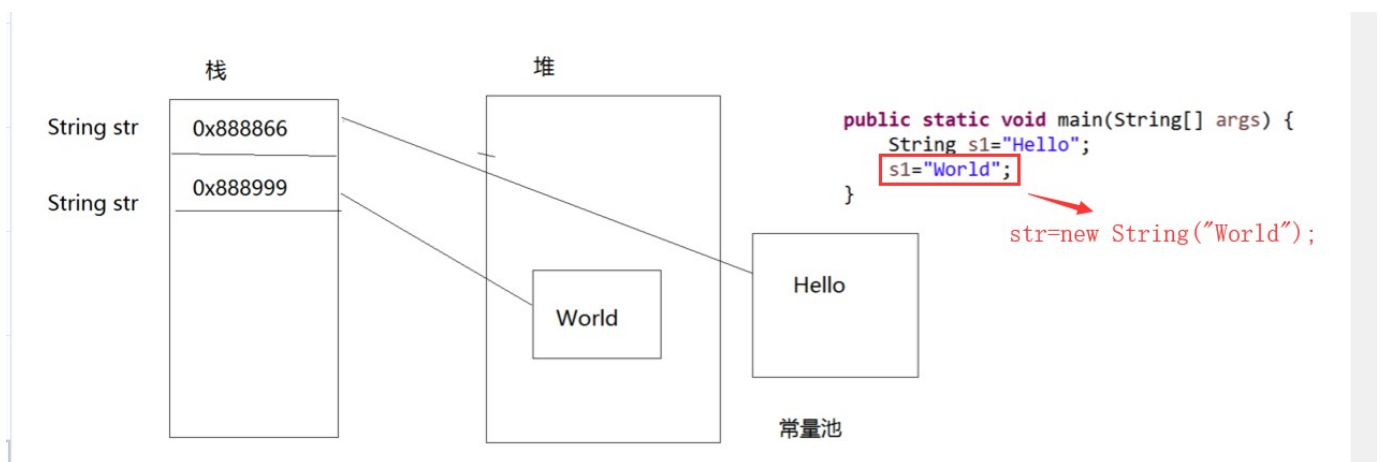
6      //    * "."不能是第一个
7      //    * 判断"."之后的字符串是不是"java"
8      int index=fileName.lastIndexOf(".");
9      String suffix=fileName.substring(index+1, fileName.length());
10     if(index!=-1 && index!=0 && "java".equals(suffix)) {
11         System.out.println("文件合法");
12     }else {
13         System.out.println("文件不合法");
14     }
15 }

```

* 能够理解字符串常量池

* 字符串可以用两种方式赋值,有一个非常重要的特征,即不可变性 (immutable);

* 不可变的意思是:一旦一个字符串被创建后,它的值就不能被修改;



* 为了能够重用这些不变的字符串,Java使用了字符串常量池;

* 凡是用=直接赋值的方式得到的字符串,都存储在常量池中;

* 相同的共用一个具体字符串; 使用new创建的字符串不适用常量池,每次都分配新的内存空间

```

3 public class Test11 {
4     public static void main(String[] args) {
5         String s2="Hello";
6         String s3="Hello";
7         String s4=new String("Hello");
8         String s5=new String("Hello");
9         System.out.println(s2==s3);
10        System.out.println(s2==s4);
11        System.out.println(s2==s5);
12    }
13 }
14

```

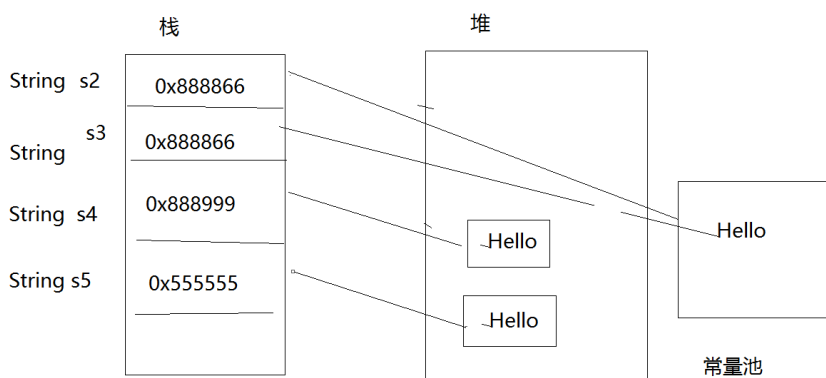
第二次从常量池里面获取

使用new 构建，不能用到常量池的技术

Problems @ Javadoc Declaration Search Console

<terminated> Test11 (2) [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (2018年8月8日 下午7:54:36)

true
false
false



```

public static void main(String[] args) {
    String s2="Hello";
    String s3="Hello";
    String s4=new String("Hello");
    String s5=new String("Hello");
}

```