复习：

1. String：操作字符串但是不改变原字符串

2. Pattern：指定规则

str.replaceAll(“(.)\\1+”, “$1”);

## 包装类

针对每一种基本类型都提供对应的类类型，然后利用这些类产生的对象来便捷的操作的数据。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| byte | short | int | long | float | double | char | boolean | void |
| Byte | Short | Integer | Long | Float | Double | Character | Boolean | Void |

注意：Void本身是一个最终类，不能创建对象

将基本类型转化为对应的引用类型的对象---封箱

将基本类型直接赋值给引用类型---自动封箱，底层实际是调用了对应类身上的valueOf方法

将引用类型直接赋值给基本类型---自动拆箱，底层实际上调用了对应对象身上的\*\*\*Value方法

自动封箱/拆箱是JDK1.5的特性之一

考虑：如何将字符串转化为对应的字符？

“c” -> ‘c’ char ch = “c”.charAt(0); char ch = “c”.toCharArray()[0];

注意：八种包装类型对应的对象的哈希码是固定不变的

## 数学类

Math---最终类，只针对基本类型提供了基本的数学运算，本身构造方法私有了，提供了大量的静态方法，本身是作为一个完全的工具类来使用的。

BigDecimal---用于精确运算的类，需要将参数以字符串形式传递

## 日期类

Date---表示日期的对象，需要掌握将字符串转化为日期的方法---SimpleDateFormat 中的parse方法，在转换的时候需要指定的转化格式

Calendar --- 仅作了解

## JDK8的部分内容

### 接口中的默认方法

从JDK1.8开始，允许在接口中定义的普通的实体方法，但是必须用default来修饰。

也允许在接口中定义静态方法

### Lambda表达式

(参数) -> {代码块};

用于去重写接口中的抽象方法。

注意：要求能够使用Lambda表达式的接口中只能有一个抽象方法

如果接口中只有一个抽象方法，这个接口称之为函数式接口

不能用Object对象来接住Lambda表达式

### 时间包

对时间体系重新划分，划分出来一个全新的时间包---java.time