# Day：Java基础（其他）

## 序列化

1、序列化指把应用层的对象或数据结构转换成一段连续的二进制串，反序列化指把二进制串转换成应用层的对象或数据结构

2、序列化按可读性可分为两类：二进制序列化、文本序列化（如 XML、JSON）以及半文本序列化。

3、文本序列化可读性、可编辑性好；二进制序列化不具有可读性，但解析速度更有优势

4、Java 中，有两种方式可以实现二进制序列化，既可以实现 Serializable 接口，也可以实现 Parcelable 接口。

5、在 Android 中，我们不应该使用 Serializable 接口。因为 Serializable 接口使用了反射机制，这个过程相对缓慢，而且往往会产生出很多临时对象，这样可能会触发垃圾回收器频繁地进行垃圾回收。相比而言，Parcelable 接口比 Serializable 接口效率更高，性能方面要高出 10x 多倍。

## Java类型转换

1、Java中的类型转换分为基本数据类型转换和引用数据类型转换

2、基本数据类型转换分为自动转换和强制转换，自动转换指从位数低的类型向位数高的类型转换，强制类型转换指从位数高的类型向位数低的类型转换，会导致精度缺失

3、引用数据类型转换时，子类可以转换成父类，父类不一定可以转换为子类，父类引用引用的是子类对象时可以强制转换为子类，但是父类引用引用的是父类对象时，强制转换会抛出异常

## 泛型和类型擦除（未完待续）

1、泛型就是指宽泛的数据类型，允许在定义类、接口和方法的时候使用类型参数（type parameter）

2、声明的类型参数在使用时用具体的类型来替换

3、使用泛型的时候加上的类型参数，会被编译器在编译的时候去掉，这个过程就称为类型擦除

## String、StringBuffer 和StringBuilder 的区别

1、String 是不可变对象，每次对 String 对象进行更改时，都会生成一个新的对象，然后将指针指向新的对象

2、StringBuffer 线程安全，使用时，每次都是直接对 StringBuffer 对象本身进行操作，并不生成新的对象

3、StringBuilder 非线程安全，和 StringBuffer 类似，单线程中性能比 StringBuffer 高

4、StringBuilder 如何避免不可变字符串分配的问题

## 自动装箱和自动拆箱

1、自动装箱指 Java 自动将原始数据类型转换为对应的对象类型，如 int 变量转为 Integer 对象；将 Integer 对象自动转换为 int 类型值，叫做自动拆箱

2、自动装箱时编译器调用 valueOf 将原始类型值转换成对象

3、自动拆箱时，编译器通过调用类似 intValue(), doubleValue() 这类的方法将对象转换成原始类型值

4、自动装箱、拆箱主要发生在赋值时、方法调用时