

java.lang.* 中

基本类型

byte

short

int

long

包装类

Byte

Short

Integer

Long

基本类型

float

double

char

boolean

包装类

Float

Double

Character

Boolean

基本数据类型 $\xrightarrow[\text{拆箱}]{\text{装箱}}$ 包装类

装箱: 构造方法: Integer (int value) (过时方法)
Integer (String s)

静态方法: static Integer valueOf (int i)
static Integer valueOf (String s)

拆箱: 成员方法: int intValue()

自动装箱/拆箱: 基本数据类型和包装类的转换

基本数据类型与字符串的转换

字符串 \rightarrow 基本数据类型

parse 方法: public static byte parseByte (String s)

(Character 类无该方法) (其余的更换红线处) (int i = Integer.parseInt(s))

(一定要匹配好输入和输出, 否则会抛出 NumberFormatException)

基本数据类型 \rightarrow 字符串

1. 基本数据类型的值 + ""

2. 静态方法: ① 包装类: static String toString (int i) 返回 int i 的字符串对象
② String 类: static String valueOf (int i)

java.lang.Object 类是java语言的根类,即所有类的父类,所有的类都继承了这个类

```
//直接打印对象的名字,其实就是调用对象的toString方法 p = p.toString();  
System.out.println(p); //com.itheima.demo01.Object.Person@50cbc42f | abc | Person{name=张三, age=18}  
  
/*  
    看一个类是否重写了toString方法,直接打印这个类对应对象的名字即可  
    如果没有重写toString方法,那么打印的就是对象的地址值(默认)  
    如果重写toString方法,那么就按照重写的方式打印  
*/  
Reader r = new Reader();
```

Object类中equals()方法默认比较对象的地址值,无意义.
故需要重写equals()方法来比较对象的内容

枚举类的对象只有确定的有限个

需要定义一组常量时,使用枚举类

枚举类的对象不能被改动,应用 private final 修饰

枚举类不需要被访问,应为 static

枚举声明包含:(1)枚举常量;(2)其他成员(如构造器、字段、方法)

在Java中提供了四种访问权限，使用不同的访问权限修饰符修饰时，被修饰的内容会有不同的访问权限，

- public：公共的。
- protected：受保护的
- default：默认的
- private：私有的

2.2 不同权限的访问能力

	public	protected	default（空的）	private
同一类中	√	√	√	√
同一包中(子类与无关类)	√	√	√	
不同包的子类	√	√		
不同包中的无关类	√			

可见，public具有最大权限。private则是最小权限。

编写代码时，如果没有特殊的考虑，建议这样使用权限：

- 成员变量使用 `private`，隐藏细节。
- 构造方法使用 `public`，方便创建对象。
- 成员方法使用 `public`，方便调用方法。

小贴士：不加权限修饰符，其访问能力与default修饰符相同

