对象：是类的一个实例，有状态和行为。

类：描述对象，一个模板

方法：行为，一个类有多种方法

实例变量：每个对象有独特的实例变量。

public static void main (String[] args)
↓ ↓ ↓ ↓ ↓ ↓
访问修饰符 关键字 返回类型 方法名 String类 字符串数组

修饰符 → 访问：default、public、protected、private
修饰符 → 非访问：final、abstract、synchronized

关键字

static 关键字的作用

1、static 用于修饰全局变量

被修饰的变量称为类变量或静态变量。所有实例共享一个静态变量，当某个实例修改了静态变量的值，这个修改会所反映到其他实例上。

2、static 不能修饰局部变量。

3、static 修饰方法。

静态方法属于类，但不属于类的任何特定实例。

可以不创建类的实例直接调用静态方法。

4、Static方法的继承关系。

静态方法不能被覆盖。

静态调用与类型相关。

5、Static修饰代码块。

修饰的代码块只能在全局作用域，无法修饰内部。

6、static修饰类。

static主要修饰内部类。

封装：对象状态私有化、通过公共方法访问。

```
private String name;
public String getName() { return name; }
```

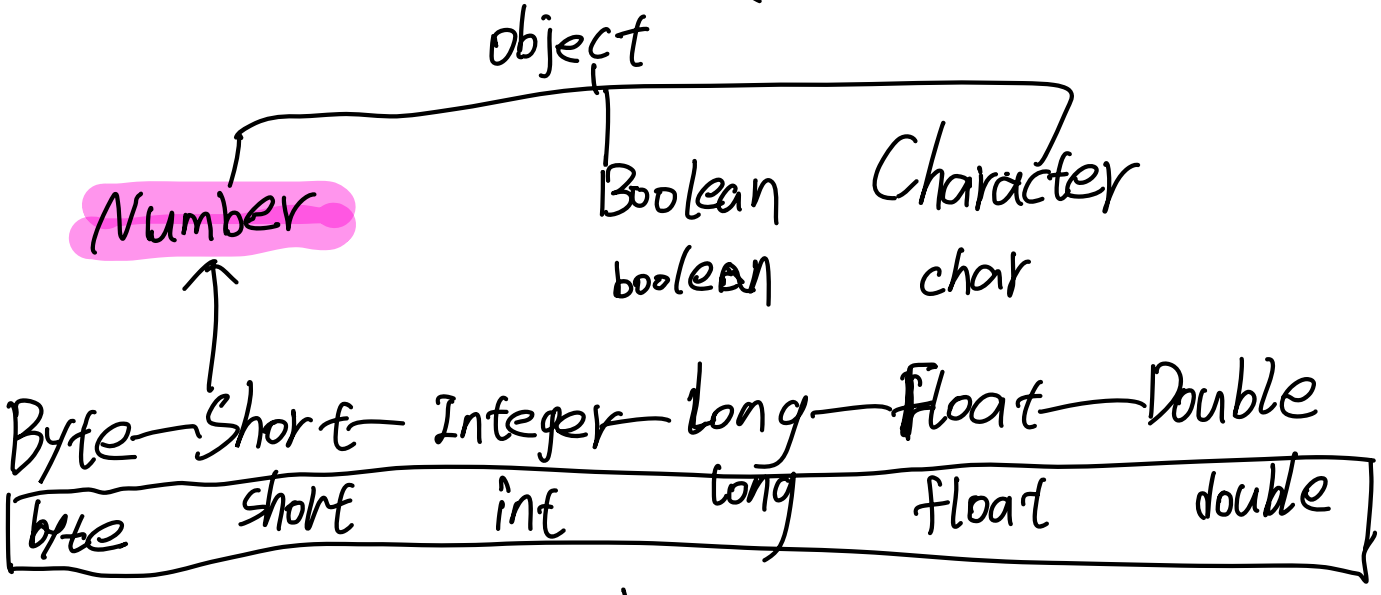
多态：对象有多种形态，通过方法的重写重载实现多态。

抽象：用抽象类、接口来定义实现的方法，不提供具体实现

抽象类：public abstract class Shape{
abstract void draw();

接口：public interface Animal {void eat();}

包装类（integer、Long、Byte、Double、Float、Short）
都是抽象类Number的子类。



int与 integer 区别

int为基本数据类型，直接存储整数值，效率高，

Integer是int的包装类，为对象类型。

内存占用和性能：
int直接存储在栈内存
integer是对象，存储在堆内存

默认值.

int → 0

Integer → null

自动装箱与拆箱

```
int a = 5;
Integer b = a; // 自动装箱
int c = b;     // 拆箱
```

比较方式

int 直接比较.

```
int a = 10, int b = 10;
Sout(a == b); // true.
```

Integer, == 比较的是引用(地址)值比较

需要 equals()

"==" 与 equals() 方法的区别.

"==" 运算符: 比较两个变量的值 (基本类型 int, double, char)

内存地址 (引用类型 对象、数组.)

equal()方法，可重写。

主要用于引用类型，比较两个对象内容是否逻辑相同，

集合和数组区别：

对比项	数组（Array）	集合（Collection）
长度	固定（不可变）	动态扩容（如 `ArrayList` 自动增长）
存储类型	基本类型 + 对象	只能存储对象（基本类型需装箱）
功能方法	仅支持 `length` 和索引访问	提供丰富方法（增删改查、遍历等）
线程安全	无（需手动同步）	部分集合是线程安全的（如 `Vector`）
性能	访问快（连续内存）	部分操作可能较慢（如 `LinkedList` 随机访问）
内存占用	更小（无额外对象开销）	更大（对象存储 + 扩容机制）
适用场景	数据量固定、高效访问（如数学计算）	数据量不确定、频繁增删（如业务逻辑）

常见集合：

List（有序，可重复）

ArrayList → 基于数组，查询快，增删慢。

LinkedList → 基于链表，增删快，查询慢。

Set（无序，唯一）

HashSet：基于HashMap，去重快。

TreeSet：红黑树，自动排序。

Map（键值对）

HashMap：哈希表实现，查找快。

TreeMap：红黑树，按键排序