主类和文件同名 对象 方法 属性 实例 Stream

什么是对象:从编程角度是变量和函数的容器，从工程角度自顶而下，是对现实内容映射到计算机的抽象建模，对象由属性和方法组成，属性值也可以是对象。任何事物都可以是对象，数值可以是对象，字符串可以是对象，数组可以是对象，窗口可以是对象，html元素可以是对象，元素属性可以是对象，游戏角色是对象。

数据类型分为基本类型和引用类型，规定了数据的范围，和数据的操作

计算机是数字组成的，指令和操作数，内存地址，字符 都是数 表达式语句是对CPU汇编操作的封装 而函数是对表达式的封装 声明，调用 形参 实参 外部变量的引用传参和值传参 孤儿对象（无引用的时候清除）

一个函数处理的数:对象的属性 传入的参数（可以是数组或者对象） 内部定义的局部变量 直接引用外部的变量 返回值 内部定义的函数 调用的外部函数/对象方法/构造函数 函数分为处理函数和返回值函数,也可以既处理数据又返回值

关于向上转型 只能使用父类中定义的方法或者接口中定义的方法

反射 注解

包和文件夹的区别

定义main(){}是在哪调用 主类何时实例化？ 和调用 定义和声明时要带类型，其他时候不用

一、对象 1、类 public class 类{} 成员变量（属性） 成员方法 变量类型 方法类型是返回值类型 2、构造方法，和类名相同 this指向实例对象 3、对象实例化 （1）声明类型 引用变量（2）实例化 new 赋值给引用对象 4、访问对象属性和方法 5、包 import

This引用对象属性时，在没有同名参数时，是不必写this的

二、数据类型 字节byte 8 短整 16 short 整型 32 int 长整64 long 浮点 32 双精度 64double 布尔 boolern 字符char 16 2、包装类型 静态属性 长度 最大值 最小值

3、引用类型 指向对象 类型不可变

4、常量 final 不可修改 5、转义 \n \r \f \b \0 \s \t \" \' \\

5、自动类型转换（混合运算）

6、强制类型转换 (类型)变量 然后赋值给新变量

三、变量 局部 对象属性 静态变量（类变量） 2、默认值 3、堆栈生命周期

四、修饰符 1、访问修饰 默认default 同一包 私有private同一类 保护protected子类 公开public 2、修饰 方法 类 变量 3、静态 4、最终final 不能引用其他对象 方法不能被改写 类不能继承 5、抽象 6、 synchronized transient volatile

五、运算符 1、算术 2、关系 3、位 4、逻辑 5、赋值 6、其他

六、循环 while for break continue

For ( item :items ) foreach for(: )

七、分支

Switch(){

Case : break;

}

八、数字包装类 数字类和子类整数类，浮点类 生成数字对象 1、装箱 基础数据赋值给包装变量 计算拆箱

2、包装对象方法 类型转换.xxxValue() 比较.compareTo(数2) 是否相等.equals(数2) 返回基础类型 .ValueOf() 返回字符串.toString() 字符串转为Int .parseInt() 3、数学类Math 基本都是静态方法 三角 .sin() 随机数.random()

九、字符包装类 1、Character c=new Character('') 2、装箱 c='' 3、转义 4、方法 是否字母.isLetter() 是否数字（） 是否空格 是否大写 转为大写 返回字符串

十、字符串类 1、字符串对象String s=""

字符数组转字符串 new String(charArray)

2、构造方法 3、长度 .length() 4、连接字符串 .concat("") + 5、格式化

6、方法 按索引找字符 比较 存储为byte数组 存储为字符数组 字符索引

正则 匹配 替换 拆分 截取子串 大写 小写 清除空格

十一、字符串缓冲和字符串构建StringBuffer

2、StringBuffer方法

添加 .append 反转 reverse 移除子串delete 指定位置插入子串insert 替换子串

其他方法类似于string

3、StringBuilder方法

十二、数组（JAVA数组是静态类型） 1、声明数组变量类型 类型[] 变量名 2、 创建数组 new 类型[长度] 3、单个元素赋值 4、遍历 循环和foreach 5、数组作为参数:类型[] 数组名,在函数内部使用数组 6、数组作为返回值 7、多维 8、数组类。都是静态的 Arrays. 排序.sort(组) 填充每个元素 .fill(组，值) 比较 .equals(组1,组2) 二分查找

十三、日期时间

十四、正则

十五、方法

十六、流、文件、IO

1. 控制台输入

输入System.in 缓存读取 字符流 字符串

2、输出 System.out.print() System.out引用了PrintStream 的对象

3、读写文件

字符流 Reader Writer 类型 缓冲流 字符串 管道 字符数组 文件

字节流 InputStream OutputSteam File

New Reader或者Stream 传什么参数 File传地址

方法read() write() 写入到输出流中 即文件中?

Writer.append()

流本身可以作为参数输入到流中

3、File 创建文件夹 读取目录 判断是文件还是文件夹 isDirectory list（） 删除目录或文件

分别用输入流和输出流打开文件 ，功能是相反的

字符流用于读写文本 字节流用于读写二进制文件 二进制文件可以用字节数组保存。

十七、输入Scanner

Scanner s = new Scanner(System.in);

十八、异常处理

检查性异常 这种异常必须捕获处理，也就是必须try catch 方法 throws

比如文件io就必须检查 throws的作用 是声明 可能会出现异常

1. 异常类 catch来捕捉

类型 异常运算 非法索引 对象数组类型错误 错误参数

找不到类 变量不存在 方法不存在 无法克隆 线程中断错误

1. 异常方法 catch中执行 getMessage printStackTrace

十九、继承 1、class 子类 extends 父类{} 2、私有方法属性不能继承，final类不能继承

3、抽象类 含有抽象方法的类（没有方法体，只有类型和参数） 4、接口，方法都没有体的类，用于多继承 interface {} 5、接口实现 类 implements 接口类{} 6、 访问父成员和属性 super 访问自己成员和属性 this 7、构造器 子类不继承父类 可以调用父类构造器 private属性和方法不能继承

二十、重写和重载 1、重写父类方法 类型必须相同 final static不能重写 2、调用父类方法 super.方法（）

3、重载 方法名相同，参数类型或数目不同

4、重写时可以调用父方法 super.method()

二十一、多态 1、向上转型 （1）定义父类型的引用变量 （2）创建子类实例 （3）父类变量引用子类实例 2、 会调用子类的方法，但无法调用父类不存在的方法。 3、虚方法

二十二、抽象类1、 abstrace class 2、继承抽象类

二十三、接口 1、接口全是抽象方法，接口没有成员变量？public static final

2、接口是类型约定

二十四、包 打包package 地址 最后会实现自动保存到相应地址 . . ... /////

一般会用域名的颠倒作为包名，之后的路径作为子包名，编译后的class文件将以目录结构保存

二十五、数据结构 1、枚举（淘汰） 2、位集合 3、向量 4、栈 5、字典 6、属性

二十六 集合框架 (接口、实现类) 1、接口 集合 列表(线性表)List 无序不重复列表set 有序列表 映射 Map 有序映射 SortedMap 2、实现类

抽象集合 抽象线性表 链表 可变数组 抽象Set 树Set 哈希映射 3、向量 栈 哈希表 集合算法 迭代器接口 比较器

集合遍历 比较

二十七、泛型 一个方法可以用于不同类型参数，但是只能是引用类型或者包装类型

1、参数类型声明 （1）泛型方法 <类型> 返回值类型 方法(类型 参数){ }

（2）泛型类 class 类<类型>{ 类型属性或方法 }

3、 泛型的使用 类型通配符 new List<String>（）

二十八、序列化（对象变为字符串，可以持久化） 实现接口 属性可序列化 2、序列化到文件 3、反序列化

二十九、网络编程 TCP UDP URL处理 套接字编程socket 1、URL处理

Nio java.net包

Soket

new ServerSocket(端口) getLocalPort 获取绑定端口 .accept() .setSoTimeout

.bind

New Socket(host,port)可以本地或者远程 .connect getlnetAddress（获取连接的地址） .getPort 返回输入流 返回输出流

2、InetAddress 类 静态方法 由ip获取对象 获取 ip

3、URL类 协议 端口 new URL(传入url参数) getPath getQuery getPort getHost getRef

4、建立客户端和服务端

5、发送邮件 Mail类

三十、发送邮件

三十一、多线程 线程类 线程对象 同步 通信 死锁 控制：挂起停止恢复 1、创建线程（1）可运行接口 Runable （2）继承Thread 构造（线程名）run(){ Thread.sleep(时间)} start(){若无则创建线程对象}（3）

2、(1)线程方法 开始执行线程.start() .run() 改变线程名.setName() 改变优先级 setPriority(int） 标记为用户线程或守护线程 中断 interrupt()

是否活动状态isAlive() （2）静态方法Thread. 暂停当前进程yield() 休眠当前进程.sleep() 返回当前进程引用.currentThread() 打印线程的堆栈.dumpStack()

3、状态 新建 就绪 运行 阻塞（挂起等待，同步，其他）

4、线程优先级

5、线程池

三十二、文档注释 javadoc 插入@

三十三、JAVA8、9、10的新特性

匿名方法 或者说箭头函数 Stream.api datetime optional类空指针异常 base64