代码的访问级别 class关键子后紧跟类名,保留字不能做类名 源代码的文件名必须和public类的类名相同 main方法必须声明为public ,每个java应用程序必须 一个简单的应用程序 有一个main方法 点号 (.) 用于调用方法。 函数调用格式: object . method(parameters) 声明一条完整的java语句必须以分号结束 //: 从//开始到本行结尾 /\* \*/: 将一段比较长的注释括起来,/\*\*/不能嵌套使 /\*\* \*/: 可以用来自动生成文档,以/\*\* 开始, \*/结束 子主题 4 整形4种: int (4字节) 、short (2字节) 、long (8 字节)、 byte (1字节) 浮点类型2种: float (4字节) 该类型的数值后缀F或f, 需要单精度数据的库时使用 double (8字节) 该类型的数值精度是float类型的两 倍,可带后缀D/d,默认浮点型数据为double, double 类型的数据可以不带后缀 DigDecimal类可以实现四舍五入 数据类型(8种)没有无符号形式的类型 码点: 指与一个编码表中的某个字符对应的代码值 char类型: char类型的字面量值要用单引号括起来 在基本的多语言级别中,每个字符用16位表示,通常被 ) 强烈建议不要再程序中使用char类型 称为代码单元 子主题 3 Boolean类型: false 和true 用来判断逻辑条件 类型 变量名 变量名:变量名必须是一个以字母开头并由字母或数字构成序列。但是保留 字、符号(+-空格等)不行,大小写敏感,变量名的长度基本没有限制 声明一个变量后,必须要初始化 常量 关键字final指示常量,常量名使用全大写 类常量: static final 类常量的定义必须定位于main 方法的外部 算术运算符 + -\*/(整数除法、浮点数除法) 求余 用来存储同一类型值的数据结构(集合) (求模) % strictfp 标记的类的所有方法都要使用严格的浮点计算 Math.round方法: 进行舍入运算 初始化一个匿名数组 : new 数据类型 [ ] {,,,,,}; 要得到一个完全可预测的结果比运算速度更重要的话, 用于这类似的情况: smallPrimes = new int [] 格式: 数据类型 [] 变量名 = new 构造器; math类中的方法时静态方法 🕤 应该使用 StrictMath类 数据类型 [] 变量名 = { , , , , }; Math . PI  $-\pi$ 数组的长度不会改变,当要使用超过长度的地方,会出现array index out of bounds 异常而终止 Math.E -e for each 和for 两个的区别在于for each 不须要操心 如果两个操作数中有double,另一个会转换为double 数组下标值 for each 循环: 格式为 for (variable : collection) statement 如果两个操作数中有float,另一个会转换为float collection 必须是一个数组或者一个实现了Iterable接口的类对象 要打印出数组中的所有值,可以使用Arrays类的 如果两个操作数中有long,另一个会转换为long java的基本程序设计结构 toString() 方法 数值类型间的转换 java中允许讲一个数组变量拷贝给另一个数组变量 强制转换 ⑤ 格式: (类型) 需要转换的 java中的[]运算符被预定义为检查数组边界,并且没有 结合赋值和运算符: -= 、+= 、/= 、 %= 、\*= 、 指针运算 关系和Boolean运算符: == 、! + 、、、、 命令行参数 : 在main方法中 String args [] (参数) 位运算符: &、|、^、~、>>、 <<、>>> 表明main方法将接收一个字符串数组(命令行参数) 括号与运算符 for each 循环不能自动处理二维数组的每一个元素。 它是按照行(就是按照一位数组处理的),如果想要访。多维数组 使用多个下标访问数组元素, 枚举类型: 关键字enum l格式: enum 名字 {, , , , , } 问二维数组中的所有元素,需要嵌套使用for each 声明格式和一维数组类似 不规则数组,即每一行长度不相同的数组 如: int[][] java没有内置的字符串类型,java类库中提供了一个预 = new int [i+2][]; 定义库: String 在Java中允许数组长度为0 长度为 0 与 null 是不同的 Scanner类并不适合用于控制台读取密码,java SE引入了Console类 实现了这个目的console对象没有scanner方便,它每次只能读取一行 输入,没有能够读取一个单词或一个数值的方法 Scanner 类的各种方法实现输入 ◎ 子主题 1 格式化输出 printf 方法 🏻 子主题 1 if(condition) stationment if(condion) stationment1 else stationment 2 while (condtion ) stationment; do stationment while (condtion); for(int i= 0; i<=10;i++) {, , , , } 多重选择 switch (chioce) ○ case 标签可以是: 类型是char、byte、short或int的 break; 常量表达式 枚举常量 控制流程 从Java SE7 开始,case标签还可以是字符串字面量 java提供了一种带标签的break语句,用于跳出多重嵌 套的循环语句(任何语句,甚至是if 语句和块语句 (只 能跳出块语句,不能跳入块语句)) 注意:标签必须 放在希望跳出的最外层循环之前,并且必须紧跟一个冒 continue将控制转移到最内层循环的首部 for循环中continue将控制跳转到更新部分 continue语句它是中断正常的控制流程 带标签的continue语句,将控制跳转到与标签匹配的循 环首部 当基本的整数和浮点数精度不能满足要求时,Math包有两个类: BigDecimal和BigInteger BigInteger 任意精度的整数运算

JAVA区分大小写

访问修饰符(public、):控制程序的其他部分对这段

不能使用我们熟悉的算术运算符 \* - + /。需要使用 大数据类中的方法如: add multiply