

第三题

Task1

1

八种基本数据类型

byte short int long char float double boolean

2

四种整形数据类型

byte: 1个字节, 从-128到127。

short: 2个字节, 从-32,768到32,767。

int: 4个字节, 从-2,147,483,648到2,147,483,647

long: 8个字节, 从-9,223,372,036,854,775,808到9,223,372,036,854,775,807

觉得byte和short类型有点鸡肋, 是不是上古时期内存特别小的时候的产物, 现在应该用的很少吧

自动类型转换

一个小范围的数据类型(如int)和一个大范围的数据类型(如long)参与运算, 就会自动将小范围类型提升到大范围类型进行运算。强制类型转换

优点: 确保数据按照你想要的特定方式变换(如由大范围数据类型向小范围数据类型变换, 与自动变换方向相反)

缺点: 可能会导致数据精度下降 可能会转换失败(如将一个非数字字符转换为整数)

3

自动类型转换 52

变量c 是字符(char), 其内部存储的是ASCII码值, '0' 的 ASCII 码对应的十进制数值是48

所以c不等于0, 而是等于48 所以b=4+48

4

包装类:

就是把基本类型的数据包装成对象。有八种, 对应基本数据类型。

(1) false (2) true (3) false

解释: 有两个关键点,

1是(1)和(2)的区别, 2是为什么(2)和(3)的结果不一样

1==在Java和C语言的含义不是完全一样的, 在Java中==还会检查它们是否指向同一个内存位置。因为new Integer(n)会产生一个新的对象, 所以x和y分别是两个独立的新实例, 不在同一个地址。

2 Integer 类型会有一个范围, 如果给定的数字在 Integer 的范围(-128 到 127)内并且已经有一个对象代表这个值, 那么两次调用会返回同一个对象; 但是因为mp都超出了Integer 类型的内置常量, 所以会各自创建新的对象, 导致判断结果为false。

Task2

5

13

6 8

++在前和在后是不一样的, 如果a=1,那么++a=2, a++还是1 在经历了这个之后a都=2

跟c语言是一样的吧

6

在学习和过程中,我发现要了解补码, 就应该也了解机器数和真值, 原码和反码

机器数: 一个数在计算机中的二进制表示形式, 叫做这个数的机器数

计算机字长8位, 第一位表示正负

真值: 将带符号位的机器数对应的真正数值称为机器数的真值(因为如果是整数的话第一位为1, 会把数字变大, 就不是原来的值了)

原码: 原码就是符号位加上真值的绝对值

反码: 把原码中的10倒过来

补码: 在反码的基础上+1(如果最后一位是一就不变, 最后一位不是1, 就把0变成1)

一般问某个数的二进制形式, 都采用的是补码形式, 因为使用原码和反码对0和负数的表示会产生混淆

a的二进制形式0000 0010

-a 1111 1101

按位与操作: &

操作流程: 如果两个数字都为1, 则结果位是1; 如果有一个或两者都是0, 则结果位是0

因为a与-a是相反的, 所以a & (-a)的结果是 0000 0000, 也就是二进制的 0 或者十进制的 0

补一点想法 在学习过程中, 有一些概念确实不好理解, 而在一些网课中, 都偏向于在深入了之后再讲解这些概念, 所以课程也不是特别好找, 有一些概念也是这一块, 那一块。

用ai搜索搜出来的也很抽象

但是确实, 这是一种提升很迅速的办法, 在了解过程中, 我常因为一个概念不理解, 一搜, 发现解释中的某些名词也不懂, 一了解就拓展了好多 于是我就在视频学习, 看书, 搜索网站来回跳转, 确实好的讲解有点难找