# 数据类型

## 基本数据类型的知识点:

基本数据类型的转换

大转小

小转大

示例:

|  |
| --- |
| **int** i = 128; **byte** b = (**byte**)i ;*//大转小.强制(显式)转换* **long** l = i ;*//小转大.自动(隐式)转换* |

## 引用数据类型:

数组,类(字符串),接口,枚举,注解

字符串本身是一个Java的常用类.它并不是一种特殊的引用类型.

# 运算符:

算术运算符 + - \* / %(取模/取余) ++ --

a+b 当前表示两个值运算出结果

赋值运算符 = += -= \*= /= %=

a+=b // a = a+b 将a+b的结果再给a重新赋值.

++及--代表是自增及自减的意思

a++ //a = a+1

++a //a = a+1

++在前与++在后的区别(在有赋值时,才会有区别)

示例:

|  |
| --- |
| **int** a = 3 ; **int** c = a++;*//先将a的值给c.自已本身再做自增运算* **int** b = ++a;*//a本身先进行运算,再将值赋值给b* System.***out***.println(c); System.***out***.println(b); |

逻辑运算符 | ||(或or)逻辑或 & &&(与and) 逻辑与

内容见PPT.

比较运算符 == < > <= >=

位运算符:

在程序开发的最初,其硬盘的条件太差.

为了让程序的运行速度更快.

所以采用了二进制直接运算的方式来达到程序运行的速度提升.

三目运算符:

布尔表达式:条件true时的执行语句?条件false时执行的语句.

字符串的比较需要用到String类重写Object类的equals()方法.

再用当前方法时,需要尽量的将有值的字符串放在调用者的位置.

Java运动机制

Java 执行编译好的\*.class文件.

Javac 编译命令,将源代码编译成JVM可以运行的字节码文件.

Javap –c \*.class

GUI 桌面级应用开发程序

AWT🡪swing🡪java FX

-------建议多看23种设计模式.

所有的程序都是运行在内存当中的.

# 面向对象

## 构造器:

创建一个类,如果没有提供构造器,

那么系统会提供一个默认的无参的构造器.

构造器的名与类名一致.

构造器不需要声明返回值类型.

提供给别的实例加以调用.

类在现实生活中,是将一些具有共性的客观世界存在的事物汇聚的集合称为一个类.

秘哲的手机.具体到一个实际的事物,就是对象.万物皆对象.

类是对象的模板,对象是类的实例.

通过new关键字将类的模板抽出一份实例放到堆内容当中.

类中只包含属性及方法.

属性是抽象出来的共性的属性(都拥有)

方法是抽象出来的共性的行为(不一定实现)

方法的设计:

返回值/参数

## 方法的重载:

方法名相同,方法的参数类型/个数不同

石头/剪刀/布

猜数字

自主创建武侠游戏

# 流程控制:

顺序结构:

从上到下依次执行代码;

选择结构:

通过boolean表达式来确定同级的某一部分代码执行.

循环结构:

将需要重复的代码进行执行.

循环的三要素;

定义一个变量

实现变量的自增或自减

条件满足时的中止循环条件

while

do...while...

for

学习资料下载:

<https://gitee.com/restyhap>