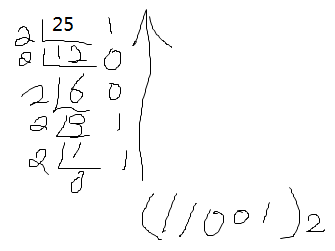
1. **Java语言释型还是编译型?还是两者都是?**

先编译再解释

编译型语言在程序执行之前，有一个单独的编译过程，将程序翻译成机器语言就不用再进行翻译了。解释型语言，是在运行的时候将程序翻译成机器语言，所以运行速度相对于编C/C++ 等都是编译型语言，而Java，C#等都是解释型语言。

1. **如何将10进制转换为2进制数据**



1. **如下写法哪些是不对的：**

**a. byte b = 30;**

**b. byte c = 500;**

**c. long d = 2343223;**

**d. float f = 3.14;**

b:不对，超出了byte的范围

d:不对，缺少f,应该写为3.14f或3.14F

byte计算时自动转换成int，所以a对

1. **浮点数能用于比较吗?下面可能打印什么结果:**

|  |
| --- |
| **float d1 = 423432423f;**  **float d2 = d1+1;**  **if(d1==d2 ){**  **System.out.println("d1==d2");**  **}else{**  **System.out.println("d1!=d2");**  **}** |

打印结果为: d1 == d2

float类型的数据在内存中的存储形式为科学计数法, 且有效位数为7位, 则d1为4.2343242E7, d2用科学计数法表示同样为4.2343242E7, 因此d1 == d2

1. **使用Scanner接收键盘输入，是否一定要加import Java.util.\*; ?**

需要导包

也可以写为import java.util.Scanner;

1. **引用类型是占用几个字节?**

引用数据类型占4个字节，用于存储对象的引用

1. **关系运算符中，能不能这么写："1<a<3"?**

不能，

应该写为a>1&&a<3

1. **运算符||和|的异同之处**

相同:

|| 与 | 都是逻辑运算符, 运算规则都为只要有一个为true,则直接返回true

不同:

|| 短路或 : 从左到右计算，如果符号左边的操作数为true,则不会继续计算,直接返回 true; 如果符号左边的操作数为false,则需要继续计算

| 逻辑或：操作数左右两边的操作数都需要计算,不会发生短路

|| 只是逻辑运算符

| 还可以是位运算符中的按位或符号

1. **5\*4最快的运算方式是?**

位运算5<<2

1. **4&5，4|5的结果分别是多少? 4&&5这个操作可行吗?为什么?**

4&5的结果为4

4｜5的结果为5

4&&5这个操作不可行，因为&&左右要求为boolean类型表达式,if后面的括号里面也只能是false或者true，不能是别的数字

1. **布尔类型能否自动转换为int? 如果不能，简述理由。**

不能

因为类型不匹配

1. **下面两种写法，哪个较好：**

**a. 70L\*60\*24\*365\*70\*20**

**b. 70\*60\*24\*365\*70\*20L**

第一种写法好;

第二种表达式的前半部分70\*60\*24\*365\*70的乘积有可能会超过int的取值范围, 导致精度损失

1. **Math.random()是什么意思?如果想获得15-20之间的随机数，怎么办?**

产生一个0-1之间的随机数，包含0但不包含1

(int)(Math.random()\*6)+15;

1. **switch中的表达式的结果需要是什么类型?**

byte,char,short,int

jdk1.5 加入了枚举

jdk1.7加入了String

1. **Java中，参数的传递使用值传递还是引用传递?**

值传递

1. **什么是方法的重载，有什么作用？**

方法重载:

[1]在同一个类中

[2]方法的名称相同

[3]参数列表不同

[4]与访问修饰符和返回值类型无关

[5]与异常无关

作用:

传递不同的参数实现相同的效果

1. **两同三不同指的是?返回值不同构成重载吗?形参名称不同构成重载吗?**

两同：同一个类中，方法名称相同

三不同：参数列表的参数的类型，个数，顺序不同

1. **【上机】使用递归算法完成阶乘算法**

public static int jiCheng(int number){

if (number==1) {

return 1;

}else{

return number\*jiCheng(number-1);

}

}

1. **栈的特点是?存放什么内容?**

栈的特点：先进后出

在函数中定义的一些基本类型的变量和对象的引用变量都在函数的栈内存中分配。

1. **堆得特点是?存放什么内容?**

**堆内存的特点：**

堆内存有内存地址，都是将内存的地址赋值给引用变量

堆内存变量无用后由垃圾回收机制不定时回收

堆内存会自动初始化

堆内存存放引用数据类，及new出来的都在堆内存里面

1. **【上机】画出如下程序的内存结构（课堂上的代码）：**

|  |
| --- |
| **//测试类和对象**  **public class TestObject{**  **public static void main(String[] args){**  **Car c1 = new Car();**  **c1.changeColor("红色");**  **c1.showColor();**  **System.out.println(Car.tyreNum);**  **System.out.println(c1.tyreNum);**    **Car c2 = new Car();**  **Engine e = new Engine();**  **e.speed = 1000;**  **e.weight = 10;**  **c2.engine = e;**  **c2.color="黑色";**  **c2.tyreNum = 10;**  **System.out.println(c1.tyreNum);**  **}**  **}**  **class Car {**  **static int tyreNum=4;**  **Engine engine;**  **String color; //char sequence :字符序列**    **void changeColor(String c){**  **color = c;**  **}**    **void showColor(){**  **System.out.println("我的颜色是："+color);**  **}**  **}**  **class Engine{**  **int speed;**  **int weight;**  **}** |

|  |
| --- |
|  |

1. **import Java.util.\*; 会不会降低程序运行速度?为什么?**

不会降低程序的运行速度

因为是导入该包下的所有的类，会降低编译速度

1. **import static 静态导入的作用是导入类还是导入类的静态属性和静态方法**

导入类的静态属性和静态方法183