晚上的作业

1 什么是类，对象，他们的关系是怎么样的。

答：类:类就是具备某些共同特征的实体的集合，它是一种抽象的数据类型，它是对所具有相同特征实体的抽象

对象:是类的一个实例，具有状态和行为

关系，对象是类实例化出来的,对象中含有类的属性,类是对象的抽象

类是具备某些共同特征的实体的集合，它是一种抽象的数据类型，它是对所具有的相同特征的实体的抽象。

2 构造方法有几种？请简单说一下

答：两种，一种无参数默认的，一种有参数，写有参数的构造方法时候，必须显式写出无参构造方法

3 数据类型有几种？以及每种的默认值是多少?请简单说一下

答：

数据类型有八种，

byte-0，short-0,int-0,long-0L

float-0.0f,double-0.0d,char-'u0000'

String-null,boolean-false

4 强制类型转换需要注意什么？

答：不能对boolean类型进行转换，对象类型不能转成不相关对象，大转小必须使用强转，转换可能会损失精度，浮点到整数转换是舍弃小数得到

5 变量有几种？他们的作用范围是什么？

答：

类变量-独立于方法之外，用static修饰，

实例变量-独立于方法之外，没有static修饰，局部变量-类方法中的变量

6 访问修饰符有几种？他们的作用范围是什么？

答： default (即默认，什么也不写）: 在同一包内可见，不使用任何修饰符。使用对象：

类、接口、变量、方法。

private : 在同一类内可见。使用对象：变量、方法。 注意：不能修饰类（外部类）

public : 对所有类可见。使用对象：类、接口、变量、方法

protected : 对同一包内的类和所有子类可见。使用对象：变量、方法。 注意：不能

修饰类（外部类）

7 static 修饰的类，变量，方法和普通对象的类，变量，方法有什么区别

答：

静态类：

全局唯一，任何一次的修改都是全局性的影响

只加载一次，优先于非静态

使用方式上不依赖于实例对象。生命周期属于类级别，从JVM 加载开始到JVM卸载结束。

静态变量：

static 关键字用来声明独立于对象的静态变量，无论一个类实例化多少对象，它的静态变量只有

一份拷贝。 静态变量也被称为类变量。局部变量不能被声明为 static 变量。

静态方法：

static 关键字用来声明独立于对象的静态方法。静态方法不能使用类的非静态变量。静态方法从

参数列表得到数据，然后计算这些数据。

8 a++ 和 ++a的区别是什么

答： a++是先赋值给a再进行增加，++a则是先增加再赋值

9 运算符优先级最高的是哪个？

答： 后缀运算符

10 循环语句有几种？请简单列一下

答：while( 布尔表达式 ) {

//循环内容

}

do {

//代码语句

}while(布尔表达式);

for(初始化; 布尔表达式; 更新) {

//代码语句

}

for(元素类型t 元素变量x : 遍历对象obj){

引用了x的java语句;

}

11 if条件语句可以有多个else吗？

答：不可以，一个if对应一个else

12 continue 和 break 在循环语句中的区别？

答：break直接跳出整个循环体，而continue只是终止本次循环进入下一次循环

13 StringBuilder 与StringBuffer的区别？

答：StringBuilder 线程安全，而StringBuffer不安全

14 final, finally的区别？

答：。被final修饰的类，就意味着不能再派生出新的子类，不能作为父类而被子类继承；finally是在异常处理时提供finally块来执行任何清除操作。不管有没有异常被抛出、捕获，finally块都会被执行

15 jvm由哪几部分组成

答：类加载器（ClassLoader）

运行时数据区（Runtime Data Area）

执行引擎（Execution Engine）

本地库接口（Native Interface）

16 用java语言描述你周围常见的事物？例如一只猫，一条狗，一座大楼，空调，手机，等等。

答：

**public** **class** Catdemo {

**private** String name;

**private** **int** age;

**public** **void** bark() {

System.***out***.println("喵喵");

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

@Override

**public** String toString() {

**return** "Catdemo [name=" + name + ", age=" + age + "]";

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Catdemo cat=**new** Catdemo();

cat.setAge(2);

cat.setName("小白");

cat.bark();

System.***out***.println(cat.toString());

}

}

17 自己手动画一下jvm的内存分部图。

