1.static变量和实例变量的区别?

1):所属不同

静态变量属于类，所以也称为类变量

成员变量属于对象，所以也称为实例变量(对象变量)

2):内存中位置不同

静态变量存储于方法区的静态区

成员变量存储于堆内存

3):内存出现时间不同

静态变量随着类的加载而加载，随着类的消失而消失

成员变量随着对象的创建而存在，随着对象的消失而消失

4):调用不同

静态变量可以通过类名调用，也可以通过对象调用

成员变量只能通过对象名调用

2.final修饰符的作用?

final修饰类：该类不能被继承

final修饰方法：方法不能被重写

final修饰变量：该变量的值不能被改变。

3.abstract修饰符的作用?

1.由abstract修饰的类叫抽象类

2.由abstract修饰的方法叫抽象方法，该方法没有方法体

Animal--abstract

+legs:int

+Animal(legs:int)

+walk() --abstract

+eat()--abstract

任务1：创建Animal类，它是所有动物类的抽象超类

任务2：创建Snake类，继承Animal类 并根据实际情况去完善

任务3：创建Cat类，继承Animal类 并根据实际情况去完善

任务4：创建Fish类，继承Animal类 并根据实际情况去完善Snake类

//加载父类静态成员 子类的静态属性

//加载父类非静态属性 调用父类的构造器

//加载子类的非静态属性 调用子类的构造器

分析运行结果

public class A {

static {

System.out.println("A 静态代码块");

}

private static String staticStr = getStaticStr();

private String str = getStr();

{

System.out.println("A 代码块");

}

public A() {

System.out.println("A 构造方法");

}

private static String getStaticStr() {

System.out.println("A 静态方法");

return null;

}

private String getStr() {

System.out.println("A 实例属性初始化");

return null;

}

public class B extends A {

private static String staticStr = getStaticStr();

static {

System.out.println("B 静态代码块");

}

public B() {

System.out.println("B 构造方法");

}

{

System.out.println("B 构造块");

}

private String str = getStr();

private static String getStaticStr() {

System.out.println("B 静态方法");

return null;

}

private String getStr() {

System.out.println("B 方法");

return null;

}

}

public static void main(String[] args) {

new B();

}

A 静态代码块

A 静态方法

B 静态方法

B 静态代码块

A 实例属性初始化

A 代码块

A 构造方法

B 构造块

B 方法

B 构造方法