# 每日反馈, 每日反馈, 一定要反馈, 我根据反馈好知道大家掌握的不好哈, 复习才有偏重点哦

周末复习一下面向对象哈

# 请补充代码，使测试类编译通过并运行出预期结果。

注意 : 以下需要自行补全代码处已使用绿色标记,请详细查看

1. 已知代码如下:

public class Demo1 {

public static void main(String[] args) {

Person p = getInstance();// 方法调用

p.eat();

}

static Person getInstance() {

// 如何补全代码能在main中输出结果 “吃饭”

return new Person();

}

static class Person {// 静态成员内部类

public void eat(){

System.out.println("吃饭");

}

}

}

运行结果：吃饭

1. 请完善下列代码，在main方法中调用Inner的method方法，并依次在控制台输出30,20,10

public class Demo3 {

public static void main(String[] args) {

// 此处需补全代码, 调用Inner中的method方法

Inner a = new Outer().new lnner();

a.method();

}

}

class Outer {

int num1 = 10;

class Inner {

int num = 20;

public void method(){

int num = 30;

// 此处需补全代码,依次输出30,20,10

System.out.println(num+”,”+super(num)+”.”+num1);

}

}

}

1. 需求：请在main方法中通过多态创建对象，随后使用对象，使得程序符合最终运行结果

public class Demo1 {

public static void main(String[] args) {

// 请在main方法中通过多态创建对象，随后使用对象，使得程序符合最终运行结果,此处需完善代码

Fu f = new Zi();

f.method();

((Zi)f).show();

}

}

class Fu {

public void method(){

System.out.println("父类的method方法");

}

}

class Zi extends Fu{

public void method(){

System.out.println("子类的method方法");

}

public void show(){

System.out.println("子类的show方法");

}

}

运行结果：

子类的method方法

子类的show方法

1. 需求：请对下列代码进行补充，打印出接口中的变量num, 随后调用method方法，method方法中代码逻辑内容为输出打印”我是method”

public class Demo4 {

public static void main(String[] args) {

// 请编写程序, 打印出接口中的变量num, 随后调用method方法,打印结果为 : 我是method ; 此处需完善代码

Iner a =new Iner(){

Public void method(){

System.out.println(“我是method:”+num);

}

}

a.method();

}

}

interface Inter {

int num = 10;

void method();

}

# 根据题目,设计程序

1. 定义动物类Animal

属性：年龄，颜色

生成空参有参构造，set和get方法

行为：eat(String something)方法(无具体行为,不同动物吃的方式和东西不一样,something表示吃的食物)

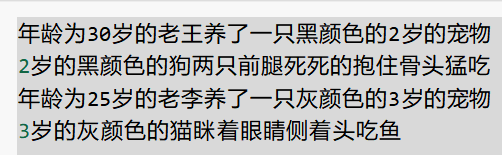
1. 定义狗类继承动物类，行为：eat(String something)方法,看家lookHome方法(无参数)
2. 定义猫类继承动物类，行为：eat(String something)方法,逮老鼠catchMouse方法(无参数)
3. 定义Person类

属性 : 姓名,年龄

生成空参有参构造,set和get方法属性:

行为：keepPet( ????? ,String something) 养宠物方法, 问号处自己思考参数类型，something表示喂养的食物

定义测试类，使用Perosn对象调用keepPet方法，实现运行结果运行结果：



提示: Person类中的keepPet方法，要求既能养猫，也能养狗，怎样设计才能同时接受两种事

# 看代码写结果

本道题目有点难, 小心分析

package com.ujiuye.test;

class BaseTmp {

BaseTmp(){//无参构造

System.out.println("BaseTmp");

}

}

class Base{

BaseTmp tmp = new BaseTmp(); //输出：BaseTmp

static {

System.out.println("Base静态"); //静态代码块 在生成字节码文件运行

}

Base(){

System.out.println("Base"); // 无参构造 运行父类无参构造 输出： BaseTmp Base

}

}

class BaseChild extends Base{

BaseTmp tmp = new BaseTmp();//成员变量 输出： BaseTmp BaseTmp Base

static {

System.out.println("BaseChild静态"); //静态代码块 在生成字节码文件运行

}

BaseChild(){

super();

System.out.println("BaseChild"); //BaseTmp Base BaseChild

}

}

class Demo{

public static void main(String[] args) {

//Base静态

· //BaseChild静态

new BaseChild();

BaseTmp Base BaseChild

}

}