# 一、选择题：

|  |  |  |
| --- | --- | --- |
| 1. | 使用权限修饰符（B）修饰的类的成员变量和成员方法，可以被当前包中所有类访问，也可以被它的子类（同一个包以及不同包中的子类）访问。（选择一项） | |
|  |  |  |
|  | A | public |
|  | B. | Protected |
|  | C. | 默认(default) |
|  | D. | Private |

|  |  |  |
| --- | --- | --- |
| 2. | 给出如下代码，如何使成员变量m被方法fun()直接访问（ C ）。（选择一项） | |
|  | class Test {  private int m;  public static void fun() {  }  } | |
|  |  |  |
|  | A | 将private int m 改为protected int m |
|  | B. | 将private int m 改为public int m |
|  | C. | 将private int m 改为static int m |
|  | D. | 将private int m 改为int m  解析：fun方法是静态方法，如果想要访问成员变量，则成员变量必须使用static进行修饰，否则编译无法通过。 |

# 二、判断题：

1、使用public修饰的成员属性和方法可以被当前项目中所有包的所有类访问。( T )

2、类的方法通常设为public，而类的实例变量一般也设为public。（ F ）

3、与未加访问控制符的缺省情况相比，public和protected修饰符扩大了类及其属性和方法的被访问范围，private修饰符则缩小了这种范围。（ T ）

4、访问权限是private的变量，只能在本类和与本类同一个包中的其他类使用。（F ）

# 三、简答题：

1、private、默认、protected、public四个权限修饰符的作用

Private:本类中访问

默认：本包中访问

Protected：本包可以直接访问，外包子类可以直接访问

Public：项目下，所用包都可以访问

2、一般属性是否要设置为private?如果属性设置为private，如何让外部访问该属性?

是，使用set()方法

3、对于boolean类型的属性，提供的getter方法是：getXXX或是isXXX?

Isxxx

4、面向对象中的封装，追求的是“高内聚，低耦合”。解释一下，内聚什么？什么叫耦合?

内聚是模块内部之间，耦合是模块和模块之间

# 四、编码题

1、使用面向对象的思想，编写自定义描述狗的信息。设定属性包括：品种，年龄，心情，名字；方法包括：叫，跑。

要求：

设置属性的私有访问权限，通过公有的get,set方法实现对属性的访问

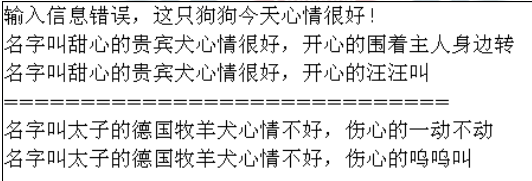
限定心情只能有“心情好”和“心情不好”两种情况，如果无效输入进行提示，默认设置“心情好”。

设置构造函数实现对属性赋值

叫和跑的方法，需要根据心情好坏，描述不同的行为方式。

编写测试类，测试狗类的对象及相关方法（测试数据信息自定义）

运行效果图:



package homework;

/\* 使用面向对象的思想，编写自定义描述狗的信息。设定属性包括：品种，年龄，心情，名字；

\* 方法包括：叫，跑。

要求：

设置属性的私有访问权限，通过公有的get,set方法实现对属性的访问

限定心情只能有“心情好”和“心情不好”两种情况，如果无效输入进行提示，默认设置“心情好”。

设置构造函数实现对属性赋值

叫和跑的方法，需要根据心情好坏，描述不同的行为方式。

编写测试类，测试狗类的对象及相关方法（测试数据信息自定义）

\*/

public class Dog {

private String breed;

private int age;

private String mood;

private String name;

public Dog(String breed, int age, String mood, String name) {

super();

this.breed = breed;

this.age = age;

this.mood = mood;

this.name = name;

}

public Dog() {

super();

}

public String getBreed() {

return breed;

}

public void setBreed(String breed) {

this.breed = breed;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

public String getMood() {

return mood;

}

public void setMood(String mood) {

if(mood.equals("心情好")||mood.equals("心情不好"))

this.mood = mood;

else

System.*out*.println("无效输入");

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

//叫和跑的方法，需要根据心情好坏，描述不同的行为方式。

public void cry() {

if(mood.equals("心情好"))

System.*out*.println("名字叫"+name+"的"+breed+"心情很好，"+"开心的旺旺叫");

else if(mood.equals("心情不好"))

System.*out*.println("名字叫"+name+"的"+breed+"心情不好，"+"伤心的呜呜叫");

else

System.*out*.println("输入信息错误，这只狗心情很好");

}

public void run() {

if(mood.equals("心情好"))

System.*out*.println("名字叫"+name+"的"+breed+"心情很好，"+"开心的围着主人身边转");

else if(mood.equals("心情不好"))

System.*out*.println("名字叫"+name+"的"+breed+"心情不好，"+"伤心的一动不动");

else

System.*out*.println("输入信息错误，这只狗心情很好");

}

public static void main(String[] args) {

//String breed, int age, String mood, String name

Dog dog = new Dog("贵宾犬",2,"心情好","甜心");

Dog dog1 = new Dog("德国牧羊犬",2,"心情不好","太子");

Dog dog3 = new Dog("德国牧羊犬",2,"1","太子");

dog.cry();

dog.run();

dog1.cry();

dog1.run();

dog3.cry();

dog3.run();

}

}

2、以面向对象的思想，编写自定义类描述IT从业者。设定属性包括：姓名，年龄，技术方向，工作年限；方法包括：工作

要求：

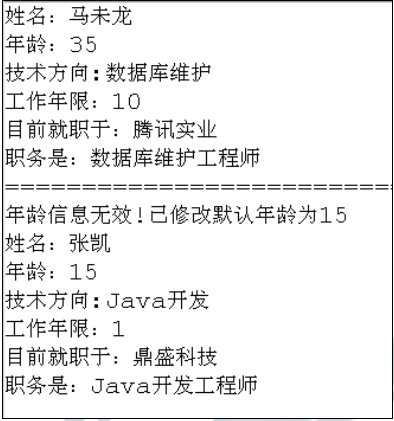
设置属性的私有访问权限，通过公有的get,set方法实现对属性的访问

限定IT从业人员必须年满15岁，无效信息需提示，并设置默认年龄为15。

工作方法通过输入参数，接收工作单位和职务，输出个人工作信息

编写测试类，测试IT从业者类的对象及相关方法（测试数据信息自定义）

运行效果图:



package homework;

import java.util.Scanner;

/\*以面向对象的思想，编写自定义类描述IT从业者。

\* 设定属性包括：姓名，年龄，技术方向，工作年限；

\* 方法包括：工作

要求：

设置属性的私有访问权限，通过公有的get,set方法实现对属性的访问

限定IT从业人员必须年满15岁，无效信息需提示，并设置默认年龄为15。

工作方法通过输入参数，接收工作单位和职务，输出个人工作信息

编写测试类，测试IT从业者类的对象及相关方法（测试数据信息自定义）

运行效果图:

\*/

public class It {

Scanner sc= new Scanner(System.*in*);

private String name;

private int age;

private String technologyTo;

private int year;

public It(String name, int age, String technologyTo, int year) {

super();

this.name = name;

this.age = age;

this.technologyTo = technologyTo;

this.year = year;

}

public It() {

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getAge() {

return age;

}

public void setAge(int age) {

//限定IT从业人员必须年满15岁，无效信息需提示，并设置默认年龄为15。

if(age>=15)

this.age = age;

else {

System.*out*.println("无效信息");

}

}

public String getTechnologyTo() {

return technologyTo;

}

public void setTechnologyTo(String technologyTo) {

this.technologyTo = technologyTo;

}

public int getYear() {

return year;

}

public void setYear(int year) {

this.year = year;

}

//工作方法通过输入参数，接收工作单位和职务，输出个人工作信息

public void work() {

System.*out*.println("目前就职：");

String workIn=sc.next();

System.*out*.println("职务是：");

String duty=sc.next();

if(age>=15) {

System.*out*.println("姓名："+name+",年龄："+age+",技术方向："+technologyTo+",工作年限："+year+",目前就职："+workIn+",职务是："+duty);

}else {

age=15;

System.*out*.println("年龄信息无效，已修改为默认的15");

System.*out*.println("姓名："+name+",年龄："+age+",技术方向："+technologyTo+",工作年限："+year+",目前就职："+workIn+",职务是："+duty);

}

}

public static void main(String[] args) {

//String name, int age, String technologyTo, int year

It it = new It("马维龙",15,"数据库维护",10);

it.work();

It it1 = new It("张凯",10,"Java开发",1);

it1.work();

}

}

# 四、可选题

1、以面向对象的思想，编写自定义类描述图书信息。设定属性包括：书名，作者，出版社名，价格；方法包括：信息介绍

要求：

设置属性的私有访问权限，通过公有的get,set方法实现对属性的访问

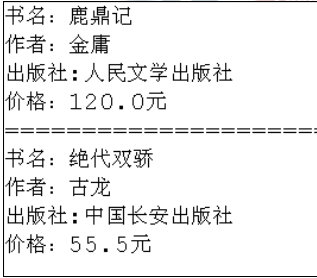
限定介格必须大于10，如果无效进行提示

设计构造方法实现对属性赋值

信息介绍方法描述图书所有信息

编写测试类，测试图书类的对象及相关方法（测试数据信息自定）

运行效果图:



2、某公司要开发名为”我爱购物狂”的购物网站，请使用面向对象的思想设计描述商品信息

要求：

分析商品类别和商品详细信息属性和方法，设计商品类别类和商品详细信息类

在商品详细信息类中通过属性描述该商品所属类别

设置属性的私有访问权限，通过公有的get,set方法实现对属性的访问

编写测试类，测试商品类别类和商品详细信息类的对象及相关方法（测试数据信息自定）

创建包info—存放商品类别类和商品详细信息类，创建包test—存放测试类

参考分析思路:

商品类别类：

属性：类别编号，类别名称

商品详细信息类：

属性：商品编号，商品名称，所属类别，商品数量（大于0），商品价格（大于0），

方法：盘点的方法，描述商品信息。内容包括商品名称，商品数量，商品价格，现在商品总价以及所属类别信息

运行效果图:

