# 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。( √ )

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

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。（ √ ）

默认访问权限 default 是pacakge访问权限

1. 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。（ × ）

Static 可以从来修饰class吗？

Private可以从来修饰class吗？

Class Outer

{

Private Static Class Inner{

}

}

Static不是说不能修饰class，而是static修饰的成员属于class。

# 三、简答题：

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

权限从高到低

1）private 只能在本类中被访问

2）默认的（访问修饰符缺省） 只能在本包中直接被访问 本包中的所有类 都可以访问到默认的访问权限；

3）protected 本包可以直接访问，外包子类可以直接访问；

4）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?

否 弄一个set方法的接口

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

GetXXX

Boolean的变量自动生成getset方法，生成的方法名是？这个再后续的一些框架处理中会出问题！！！

boolean abc;

public boolean isAbc() {

return abc;

}

public void setAbc(boolean abc) {

this.abc = abc;

}

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?

类与类之间的关系而定。高，意思是他们之间的关系要简单，明了，不要有很强的关系。类内部的方法而言。把程序的功能尽量分散，别在一个类里只写一个或很好的方法，因为那样会给你的调试等带来很多问题。出了错你都不知道在什么地方。耦合指的是对象之间的依赖性.对象之间的耦合性越高.维护成本越高.因此对象的设计应使类和构件之间的耦合最小.

# 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

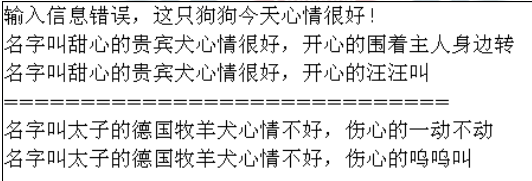
限定心情只能有“心情好”和“心情不好”两种情况，如果无效输入进行提示，默认设置“心情好”。

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 com.uek.extendsTest;

class Dog1{

private String breed;

private int age;

private String mood;

private String name;

public Dog1() {

}

public Dog1(String breed, int age, String mood, String name) {

this.breed = breed;

this.age = age;

this.mood = mood;

this.name = name;

}

public void setBreed(String breed) {

this.breed = breed;

}

public void setAge(int age) {

this.age = age;

}

public void setMood(String mood) {

this.mood = mood;

}

public void setName(String name) {

this.name = name;

}

public String getBreed() {

return breed;

}

public int getAge() {

return age;

}

public String getMood() {

return mood;

}

public String getName() {

return name;

}

public void run() {

if (mood.equals("心情好")) {

System.out.println("名字叫"+name+"的"+mood+",开心的围着主人身边转");

}

if (mood.equals("心情不好")) {

System.out.println("名字叫"+name+"的"+mood+",伤心的一动不动");

}

}

public void cry() {

if(!mood.equals("心情好")&&!mood.equals("心情不好")) {

System.out.println("输入信息错误，这只狗狗今天心情很好！");

}else if (mood.equals("心情好")) {

System.out.println("名字叫"+name+"的"+mood+",开心的汪汪叫");

}else if (mood.equals("心情不好")) {

System.out.println("名字叫"+name+"的"+mood+",伤心的呜呜叫");

}

}

}

public class zuoye {

public static void main(String[] args) {

Dog1 dog1 =new Dog1("贵宾犬",2,"心情不","甜心");

dog1.getBreed();

dog1.getAge();

dog1.getMood();

dog1.getName();

dog1.run();

dog1.cry();

Dog1 dog2 =new Dog1("贵宾犬",2,"心情好","甜心");

dog2.getBreed();

dog2.getAge();

dog2.getMood();

dog2.getName();

dog2.run();

dog2.cry();

Dog1 dog3 =new Dog1("德国牧羊犬",2,"心情不好","太子");

dog3.getBreed();

dog3.getAge();

dog3.getMood();

dog3.getName();

dog3.run();

dog3.cry();

}

}

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：

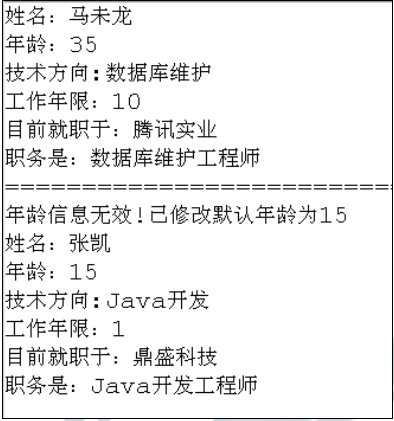
设置属性的私有访问权限，通过公有的get,set方法实现对属性的访问

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 com.uek.extendsTest;

class Itworker{

private String name;

private int age = 15;

private String technicalDirections;

private int workExperience;

public Itworker() {

}

public Itworker(String name, int age, String technicalDirections, int workExperience) {

this.name = name;

this.age = age;

this.technicalDirections = technicalDirections;

this.workExperience = workExperience;

}

public void setName(String name) {

this.name = name;

}

public void setAge(int age) {

if(age<15) {

System.out.println("年龄信息无效！已修改默认年龄为15");

}else {

this.age = age;

}

}

public void setTechnicalDirections(String technicalDirections) {

this.technicalDirections = technicalDirections;

}

public void setWorkExperience(int workExperience) {

this.workExperience = workExperience;

}

public int getAge() {

return age;

}

public String getName() {

return name;

}

public String getTechnicalDirections() {

return technicalDirections;

}

public int getWorkExperience() {

return workExperience;

}

public void work(String workplace,String position) {

System.out.println("姓名:"+name);

System.out.println("年龄:"+age);

System.out.println("技术方向:"+technicalDirections);

System.out.println("工作年限:"+workplace);

System.out.println("目前就职于:"+workplace+"\n"+"职务是:"+position);

}

}

public class zuoye1 {

public static void main(String[] args) {

Itworker i1=new Itworker();

i1.setName("马未龙");

i1.setAge(35);

i1.setTechnicalDirections("数据库维护");

i1.work("腾讯实业", "数据库维护工程师");

Itworker i2=new Itworker();

i2.setName("张凯");

i2.setAge(5);

i2.setTechnicalDirections("Java开发");

i2.work("鼎盛科技", "Java开发工程师");

}

}

# 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

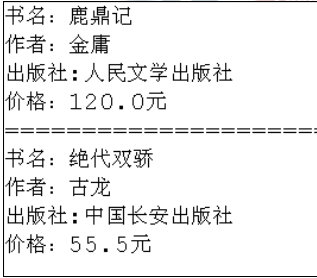
限定介格必须大于10，如果无效进行提示

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:



class Book{

private String name;

private String author;

private String press;

private double price;

public Book() {

}

public Book(String name, String author, String press, double price) {

super();

this.name = name;

this.author = author;

this.press = press;

this.price = price;

}

public void setName(String name) {

this.name = name;

}

public void setAuthor(String author) {

this.author = author;

}

public void setPress(String press) {

this.press = press;

}

public void setPrice(double price) {

this.price = price;

}

public String getName() {

return name;

}

public String getAuthor() {

return author;

}

public String getPress() {

return press;

}

public double getPrice() {

return price;

}

public void info() {

if(price<10.0) {

System.out.println("输入无效信息");

}else {

System.out.println("书名:"+name);

System.out.println("作者:"+author);

System.out.println("出版社:"+press);

System.out.println("价格:"+price);

}

}

}

public class zuoye2 {

public static void main(String[] args) {

Book b1 =new Book("鹿鼎记","金庸","人民文学出版社",120.0);

b1.info();

Book b2 =new Book();

System.out.println("====================");

b2.setName("绝代双骄");

b2.setAuthor("古龙");

b2.setPress("中国长安出版社");

b2.setPrice(55.5);

b2.info();

System.out.println("====================");

Book b3 =new Book("鹿鼎记","金庸","人民文学出版社",1.0);

b3.info();

}

}

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:

