Chp15 反射

Key Point

- Class 对象及其基本操作
- Method 对象以及 invoke 方法
- 标注

练习

1.（类对象)要获得类对象,有三种不同的方式,分别为_____、_____、_____。

说明下面几种情况，应该采用哪种获得类对象的方法

1） 获得 String 类的类对象

2） 加载一个类，该类的全限定名写在某个配置文件中

3） 打印某个对象的所有方法

2. （类对象）对类对象的说法，以下正确的是：

A. 通过类对象可以创建类的对象

B. 通过类对象，可以获得该类的私有方法

C. 通过类对象，可以获得父类的方法。

D. 同一个类的多个对象(例如多个 Student 对象),对它们调用 getClass 方法，返回的是同一个类对象（即：stu1.getClass() == stu2.getClass()）

3. （类对象）有如下代码

```
class Super{
    public void m1(){
        System.out.println(“super m1”);
    }
}
class Sub extends Super{
    public void m2(){
        System.out.println(“sub m2”)
    }
    private void m3(){
        System.out.println(“sub m3”);
    }
}
```

对 Sub 类的类对象调用 getMethods()方法，会得到_____；调用 getDeclaredMethods() 方法，会得到_____

4. （Method，重载）把下面代码补充完整

```
class MyObject{
    public void m(){}
    public void m(int i){}
    public void m(Integer i, Double d){}
}
public class MyObject{
    public static void main(String args[]) throws Exception{
        Class c = MyObject.class;
        //m1 表示 public void m()方法
        Method m1 = ____________________;
        //m2 表示 public void m(int i)方法
        Method m2 = ____________________;
        //m3 表示 public void m(Integer i, Double d)方法
        Method m3 = ____________________;
    }
}
```

5. （标注，Override）

有如下代码：

```
public class Super{
    public void m(){}
}
public class Sub extends Super{
    //1
    public void m(int n){}
}
```

问：代码是否能编译通过？如果在//1 处加上@Override，则能否编译通过？

6. *（Method）代码填空

```
package corejava.chp15;
class Worker {
    private String name;
    private int age;
    private double salary;
    //构造方法
    …
    //get/set 方法
    …
    public void work(){
        System.out.println(name + “ is working”);
    }
    public void work(int hour){
        System.out.println(name + “ is working for ”
```

```
        + hour + " hours");
    }
}
public class TestStudent{
    public static void main(String args[]) throws Exception{
        Class c;
        //利用 Class.forName 获得 Worker 类的类对象
        _____________________
        //利用类对象，获得一个 Worker 类型的对象
        Worker w = _______________;
        //利用 Worker 的 set 方法，设置 w 对象的属性：
        // name : Tom
        // age : 18
        // salary : 2500
        //调用 w 对象无参的 work 方法
        //调用 w 对象有参的 work 方法，参数为 8
    }
}
```

7. *（Method）利用反射，改写下面代码

```
    //-----------------改写下面代码-----------
    List list = new ArrayList();
    list.add("hello");
    list.add("world");
    String str = list.get(0);
    //-------------------------------------
    System.out.println(str);
```

8. **（反射）完成下面功能

已知有接口定义如下：

```
public interface Person{
```

```
    void setName(String name);
    void setAge(int age);
    void work();
}
```

1） 为Person 创建两个实现类，一个Teacher 表示老师，一个Cook 表示厨师。在Teacher 类的work 方法中输出”Teacher teach”，在Cook 的work 方法中输出”Cook make meals”。

2） 创建一个配置文件，格式如下：

```
Teacher
tom
18
```

第一行表示相应的实现类；
第二行表示相应的名字
第三行表示相应对象的年龄

3） 读取配置文件，根据配置文件的信息，创建相应的对象，并调用对象的work 方法。

9. ***（反射综合）读入一个对象信息，并利用反射，把该对象的信息按照特定格式拼成一个字符串。

例如，有如下类定义

```
class Address{
    private String addressName;
    private String zipCode;
    //get/set 方法
    //构造方法
}
class Student{
    private String name;
    private int age;
    private int score;
    private Address addr;
    //构造方法
    //get/set 方法
    public String getLevel(){
        if (score > 60) return “合格”;
        else return “不合格”;
    }
}
```

注意，在Student 中并没有定义名字叫level 的实例变量。

假设创建了如下对象：

```
Address addr = new Address(“beijing”, “100000”);
Student stu = new Student(“tom”, 18, 75, addr);
```

要求：写一个方法： public static String toJson(Object o)，如果对stu

调用 toJson 方法，则返回值为：｛“name”:”tom”, “age”:”18”, “score”:”75”,｛“addressName”:”beijing”, “zipCode”:”100000”｝, “level”:”合格”｝
提示：
1. 调用所有 Student 中的 getXXX 方法。每个 getXXX 方法意味着一个属性，其中 XXX 为属性名，而 getXXX 方法返回值为属性值。
2. 注意：Student 类中一定有一个 getClass 方法，该方法表示返回类对象，而不能作为 Student 的属性。即：不需要调用 Student 类中的 getClass 方法
3. 如果 getXXX 返回值是不是基本类型或字符串（即，返回值是一个对象，如上面的 getAddress），则需要递归的调用 toJson 方法。
4. 获得一个方法的返回值利用 Method 类的 getReturnType()方法，判断返回类型是否是基本类型利用 Class 类中的 isPrimitive()方法。