28 反射_常用方法

JAVAEE高级

一：常用方法演示

> 1、创建一个实体类

```
public class MyPersonText {
     private String name;
     private  int  age;
     public String gender;
     public int gen;

     //不同参数的构造方法
    public MyPersonText(String name, int age, String gender, int gen) {
        super();
        this.name = name;
        this.age = age;
        this.gender = gender;
        this.gen = gen;
    }

    public MyPersonText() {

    }

    public MyPersonText(String name) {
        super();
        this.name = name;

    }
    public MyPersonText(String name, int age) {
        super();
        this.name = name;
        this.age = age;
    }
//普通方法
    public void tell(){
```

```
32.             System.out.println(getName()+gender);
33.         }
34.         public void tell(String  names){
35.             System.out.println("姓名"+names+"今年"+age+gender);
36.         }
37.
38.         //对应的get set toString 方法
39.         }
```

1、Constructor

> 1、反射里调用构造方法

```
1.
2.     public static void main(String[] arge) throws Exception {
3.         // 实例Person
4.         Class personClass = MyPersonText.class;
5.         // 获取带两个参数（String,int）的构造方法
6.         Constructor constructor = personClass.getConstructor(String.cla
   ss, int.class);
7.         // 需要强制转换 a
8.         MyPersonText per = (MyPersonText) constructor.newInstance("小张
   ", 23);
9.         System.out.println(per);
10.        // 获取Student类带一个String参数的构造方x`法
11.        Constructor constructor1 = personClass.getConstructor(String.cl
   ass);
12.        MyPersonText per1 = (MyPersonText) constructor1.newInstance("xi
   ao");
13.        System.out.println(per1);
14.    }
```

2、Method

> 1、获取某类的所有方法

```
1.
2.     public static void main(String[] args) throws Exception {
3.         Class cl = MyPersonText.class;
```

```
        // 获取MyPersonText类的所有方法
        Method[] method = cl.getMethods();
        for (Method me : method) {
            if (me.getName().contains("tell")) {
                // 无参数的方法
                Constructor constructor = cl.getConstructor(null);
                MyPersonText p = (MyPersonText) constructor.newInstance(null);
                // 执行方法:执行 p对象的tell方法孔参数的
                me.invoke(p);
            }
        }

        //// 执行方法:执行 p对象的tell方法孔参数的
        // 有参数的方法
        Class A1 = MyPersonText.class;
        Method mt = A1.getMethod("tell", String.class);
        Constructor con = A1.getConstructor(null);
        MyPersonText we = (MyPersonText) con.newInstance(null);
        mt.invoke(we, "自行车");
    }
```

3、Field

1、获取某类的属性

```

    public static void main(String[] args) throws Exception {
        MyPersonText tell = new MyPersonText("小张", 23, "男", 56);
        MyPersonText tell2 = new MyPersonText("小张", 23, "男", 56);
            Class personClass = Class.forName("com.java.fs.MyPersonText");

            System.out.println("******获取所有的公共属性.并打印*****************");
            Field[] field1 = personClass.getFields();
            for (Field f : field1) {
                System.out.println(f.getName());
            }
            System.out.println("******获取所有属性.并打印***有private*****************");
            Field[] field2 = personClass.getDeclaredFields();
```

```
            for (Field f : field2) {
                System.out.println(f.getName());
            }
            System.out.println("*公共属性**获取**设置gender属性
**********");
            // 获取gender属性
            Field field = personClass.getField("gender");
            // 获取personClass对象的gender属性值
            String stuGender = (String) field.get(tell);
            System.out.println(stuGender);
            // 设置 personClass对象的gender属性值
            field.set(tell, "雌性");
            System.out.println(field.get(tell));

            System.out.println("*private属性**获取**设置gender属性
**********");
            // Field pfield = personClass.getField("gender");
            // 获取name属性
            Field pfield = personClass.getDeclaredField("name");
            // 由于name属性是private的，将name属性设置为可用
            pfield.setAccessible(true);
            // 获取stu的name属性值
            System.out.println(pfield.get(tell));
            // 获取stu的name属性值
            pfield.set(tell, "雄小写");
            System.out.println(pfield.get(tell));
    }
```