

```
//1.获取指定的属性
//getField(String fieldName):获取运行时类中声明为public的指定属性名为fieldName的属性
Field name = clazz.getField("name");
//2.创建运行时类的对象
Person p = (Person)clazz.newInstance();
System.out.println(p);
//3.将运行时类的指定的属性赋值
name.set(p, "Jerry");
System.out.println(p);

System.out.println();
//getDeclaredField(String fieldName):获取运行时类中指定的名为fieldName的属性
Field age = clazz.getDeclaredField("id");
//
age.setAccessible(true);
age.set(p, 10);
System.out.println(p);

Field id = clazz.getField("id");
```

```
//1.获取指定的属性
//getField(String fieldName):获取运行时类中声明为public的指定属性名为fieldName的属性
Field name = clazz.getField("name");
//2.创建运行时类的对象
Person p = (Person)clazz.newInstance();
System.out.println(p);
//3.将运行时类的指定的属性赋值
name.set(p, "Jerry");
System.out.println(p);

System.out.println();
//getDeclaredField(String fieldName):获取运行时类中指定的名为fieldName的属性
Field age = clazz.getDeclaredField("age");
//由于属性权限修饰符的限制,为了保证可以给属性赋值,需要在操作前使得此属性可被操作。
age.setAccessible(true);
age.set(p, 10);
System.out.println(p);
```

```

public void test3() throws Exception{
    Class clazz = Person.class;
    //getMethod(String methodName,Class ... params):获取运行时类中声明为publ
    Method m1 = clazz.getMethod("show");
    Person p = (Person)clazz.newInstance();
    //调用指定的方法: Object invoke(Object obj,Object ... obj)
    Object returnVal = m1.invoke(p);//我是一个人
    System.out.println(returnVal);//null

    Method m2 = clazz.getMethod("toString");
    Object returnVal1 = m2.invoke(p);
    System.out.println(returnVal1);//Person [name=null, age=0]
    //对于运行时类中静态方法的调用
    Method m3 = clazz.getMethod("info");
    m3.invoke(Person.class);

    //getDeclaredMethod(String methodName,Class ... params):获取运行时类中
    Method m4 = clazz.getDeclaredMethod("display",String.class,Integer
    m4.setAccessible(true);
    Object value = m4.invoke(p,"CHN",10);//我的国籍是: CHN
    System.out.println(value);//10
}

```

```

//调用指定的构造器
@Test
public void test3() throws Exception{
    String className = "com.atguigu.java.Person";
    Class clazz = Class.forName(className);

    Constructor cons = clazz.getDeclaredConstructor(String.class,int.c
    cons.setAccessible(true);
    Person p = (Person)cons.newInstance("罗伟",20);
    System.out.println(p);
}

```