**java获取对象属性类型、属性名称、属性值**

/\*\*

\* 根据属性名获取属性值

\* \*/

private Object getFieldValueByName(String fieldName, Object o) {

try {

String firstLetter = fieldName.substring(0, 1).toUpperCase();

String getter = "get" + firstLetter + fieldName.substring(1);

Method method = o.getClass().getMethod(getter, new Class[] {});

Object value = method.invoke(o, new Object[] {});

return value;

} catch (Exception e) {

log.error(e.getMessage(),e);

return null;

}

}

/\*\*

\* 获取属性名数组

\* \*/

private String[] getFiledName(Object o){

Field[] fields=o.getClass().getDeclaredFields();

String[] fieldNames=new String[fields.length];

for(int i=0;i<fields.length;i++){

System.out.println(fields[i].getType());

fieldNames[i]=fields[i].getName();

}

return fieldNames;

}

/\*\*

\* 获取属性类型(type)，属性名(name)，属性值(value)的map组成的list

\* \*/

private List getFiledsInfo(Object o){

Field[] fields=o.getClass().getDeclaredFields();

String[] fieldNames=new String[fields.length];

List list = new ArrayList();

Map infoMap=null;

for(int i=0;i<fields.length;i++){

infoMap = new HashMap();

infoMap.put("type", fields[i].getType().toString());

infoMap.put("name", fields[i].getName());

infoMap.put("value", getFieldValueByName(fields[i].getName(), o));

list.add(infoMap);

}

return list;

}

/\*\*

\* 获取对象的所有属性值，返回一个对象数组

\* \*/

public Object[] getFiledValues(Object o){

String[] fieldNames=this.getFiledName(o);

Object[] value=new Object[fieldNames.length];

for(int i=0;i<fieldNames.length;i++){

value[i]=this.getFieldValueByName(fieldNames[i], o);

}

return value;

}

# 获取方法

package cn.sunzn.reflectcase;

import java.lang.reflect.Method;

public class GetMethod {

public static void main(String[] args) {

getMethodInfo("java.util.HashSet");

}

/\*\*

\* 传入全类名获得对应类中所有方法名和参数名

\*/

@SuppressWarnings("rawtypes")

private static void getMethodInfo(String pkgName) {

try {

Class clazz = Class.forName(pkgName);

Method[] methods = clazz.getMethods();

for (Method method : methods) {

String methodName = method.getName();

System.out.println("方法名称:" + methodName);

Class<?>[] parameterTypes = method.getParameterTypes();

for (Class<?> clas : parameterTypes) {

String parameterName = clas.getName();

System.out.println("参数名称:" + parameterName);

}

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

}

}