第7章方法

第38条:检查参数的有效性.

// 传入空的时候 运行异常

**public** **void** priList(List<Object> priArr) {

**for** (Object object : priArr) {

System.***out***.println(object);

}

}

// 参数提前检测 抛出异常

**public** **void** priList(List<Object> priArr) **throws** NullPointerException {

**if** (**null** == priArr) {

**throw** **new** NullPointerException("数组为空");

}

**for** (Object object : priArr) {

System.***out***.println(object);

}

}

第39条必要时进行保护性拷贝

//修改people的名字的时候 打印也变化

**private** People people;

**public** **void** setPeople(People people) {

**this**.people = people;

}

**public** **void** pri() {

System.***out***.println("人名是:" + people.getName());

}

//不会随着people的属性改变而改变

**private** People people;

**public** **void** setPeople(People people) {

**this**.people = **new** People();

**this**.people.setName(people.getName());

}

**public** **void** pri() {

System.***out***.println("人名是:" + people.getName());

}

Ps : 类类型传入方法是地址传入,因此原类型属性变化是有影响的,基本数据类型则是数据传入 变化不会收到影响;

第40条谨慎设计方法签名—避免过长的参数列表

//使用繁琐

**public** **void** copy(**int** id,String name,**int** age,String work,String interest,String remark) {

**this**.id = id;

**this**.name = name;

**this**.age = age;

**this**.work = work;

**this**.interest = interest;

**this**.remark = remark;

}

//简单明了

**public** **void** copy(People people) {

**this**.id = people.getId();

**this**.name = people.getName();

**this**.age = people.getAge();

**this**.work = people.getWork();

**this**.interest = people.getInterest();

**this**.remark = people.getRemark();

}

第41条慎用重载

//重载类型判断

**public** **void** checkType(**int** date) {

System.***out***.println(date);

}

**public** **void** checkType(Integer date) {

System.***out***.println(date/1000);

}

Ps: 之前有代码利用重载代替 instanceof

第42条慎用可变参数

//主要是拦截可在编译期修改的问题 不需要在运行的时候再抛出异常?

第43条返回零长度的数组或者集合,而不是null

//有效减少代码中判断是否为空的代码量?