

测试单个类

刘 钦

南京大学软件学院

代码提交之前必须先
进行单元测试！

单元测试Unit Testing

- 针对软件设计的最小单元
 - 方法
- 正确性检验

如果你的方法是这样
的。。。。

独立，与其他方法无关

- `public String kindofTriangle (double side1, double side2, double side3)`
- `...`
- `return ...;`
- `}`

独立，与其他方法无关

- `interface Adder {`
 - `int add(int a, int b);`
 - `}`
 - `class AdderImpl implements Adder {`
 - `int add(int a, int b) {`
 - `return a + b;`
 - `}`
 - `}`
- ```
public class TestAdder {
 public void testSum() {
 Adder adder = new AdderImpl();
 assert(adder.add(1, 1) == 2);
 assert(adder.add(1, 2) == 3);
 assert(adder.add(2, 2) == 4);
 assert(adder.add(0, 0) == 0);
 assert(adder.add(-1, -2) == -3);
 assert(adder.add(-1, 1) == 0);
 assert(adder.add(1234, 988) == 2222);
 }
}
```

# 半独立，只与同一类的方法有关

---

- `class AdderImpl implements Adder {`
- `int a;`
- `int b;`
- `AdderImpl(int x, int y){`
- `a = x;`
- `b = y;`
- `int add() {`
- `return getA() + getB();`
- `}`
- `int getA(){`
- `return a;`
- `}`
- `int getB(){`
- `return b;`
- `}`
- `}`

## 先测被依赖的方法

```
public class TestAdder {
 public void testGetA(){
 Adder adder = new AdderImpl(1,1);
 assert(adder.getA() == 1);
 }
 public void testSum() {
 Adder adder = new AdderImpl(1,1);
 assert(adder.add() == 2);
 }
}
```

如果依赖于其他类的  
话呢？