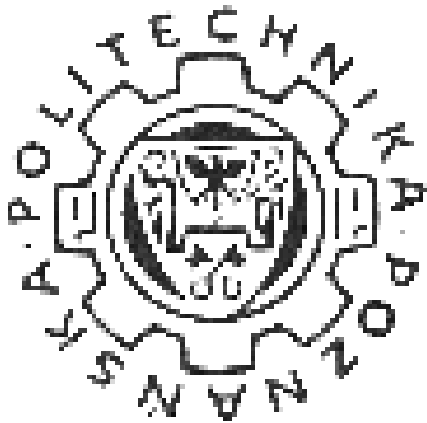




Advanced Object – Oriented Design Laboratory

Blazej.Pietrzak@cs.put.poznan
.pl

Unit

Unit – Testing private and protected methods

private

- Don't test privates (not recommended)

- Avoid private methods (recomm. to some extent)

- Use Reflection API (recommended)

protected

- Put tests in the same package as tested class (recommended)

- Use Reflection API (not recommended)

Unit – Testing with Reflection API

```
protected Object invokeMethod(  
    Object obj, String name, Object[] params)  
    throws SecurityException, NoSuchMethodException,  
    IllegalArgumentException, IllegalAccessException,  
        InvocationTargetException {  
  
    throwOnNull(obj, name, params);  
  
    Method method = getClass(obj).getDeclaredMethod(name,  
        getClasses(params));  
  
    method.setAccessible(true);  
    return method.invoke(obj, params);  
}
```

Unit – Testing with Reflection API – cont.

```
private Class getClass(Object obj) { return obj.getClass(); }
```

```
private Class[] getClasses(Object[] params) {  
    Class[] classParams = new Class[params.length];  
    for (int i = 0; i < params.length; i++) {  
        classParams[i] = params[i].getClass();  
    }  
    return classParams;  
}
```

```
private void throwOnNull(Object obj, String name, Object[] params){  
    if ((params == null) || (name == null) || (obj == null)) {  
        throw new IllegalArgumentException("null");  
    }  
}
```