



A mocking Framework for Java  
<http://mockito.org/>

by  
Petar Petrov

# Classic vs. Interaction Testing

- **state testing**

**Hey, object... How old are you?**

```
public void testAge() {  
    TestCase.assertEquals(20, person.getAge());  
}
```

- **interaction testing**

**Hey, object... What did you just do to that *other object* ?**

# What are Mocks?

# What are Mocks?

Substitutes of **real objects**...

# What are Mocks?

Substitutes of **real objects**...

... for **testing**

# What are Mocks?

Substitutes of **real objects**...



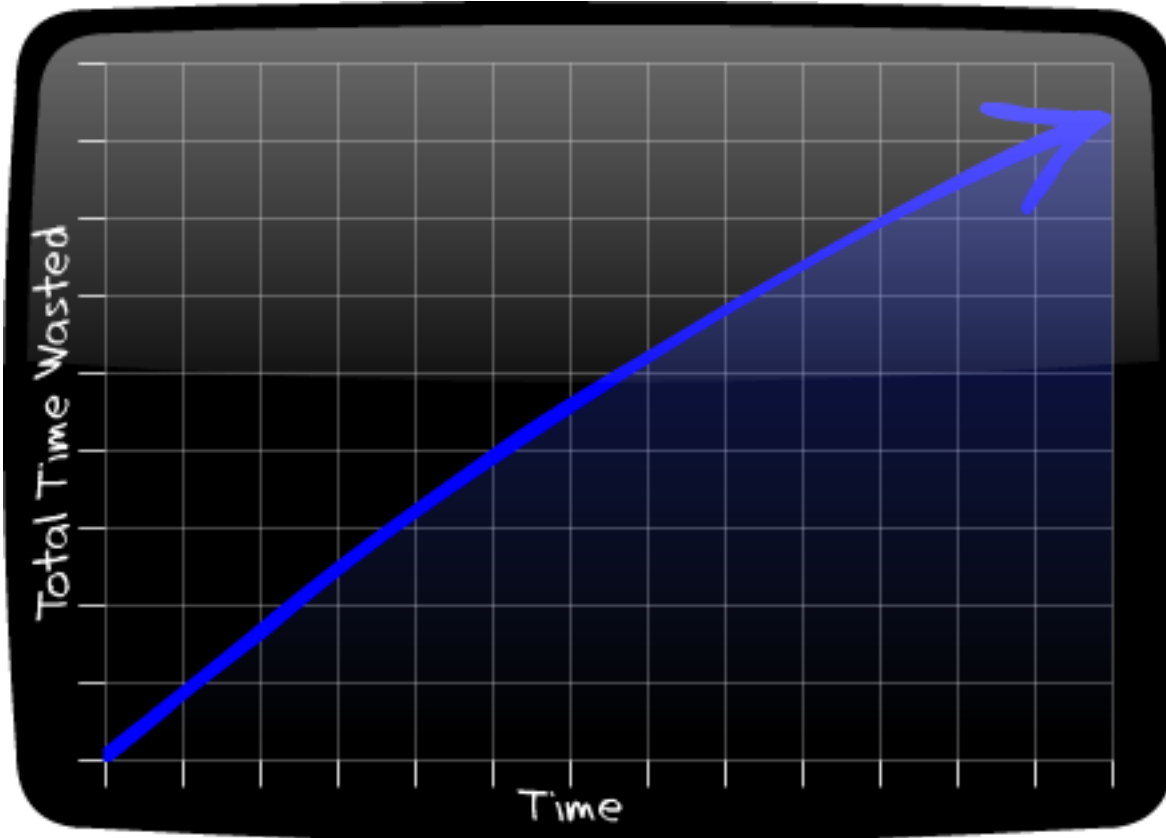
... for **testing**

# Why do we need mocks?

- Object oriented code is a web of interacting objects
- Some objects are nice and lightweight
  - String();
- Some objects are heavier
  - WholeInternetDownloader();

# Mockito in action!

Less **talking** - more **action**...



Time you wasted reading this graph, over time.



# Limitations

- Cannot mock final classes
- Cannot mock final methods - their normal behavior is executed
- Cannot mock static methods
- Cannot mock equals(), hashCode() - Mockito defines and depends upon a specific implementation of these methods. Redefining them might break Mockito.

# Warning

Because it is so easy to mock objects with Mockito, it could lead to **overmocking**

A bit **inconsistent** API

```
verify(mock).method();  
when(mock.method()).thenReturn(x);
```

# Links

<http://mockito.org/> - **official home page**

<http://monkeyisland.pl> - **official authors blog**

