Phexample

Good examples expand on one another

So should tests.

twitter.com/nesl983
testing framework
readable syntax
Stack isEmpty should be true.

This is real code!
JUnit:

```java
assertEquals(
    stack.isEmpty(),
    true
);
```

Phexample:

```
stack isEmpty should be true
```
• Examples most useful when one example expands on the previous
Two examples of a stack that expand on one another:
First example: push on a stack

Empty stack:
First example: push

StackExample>>shouldPush
| stack |

stack := Stack new
First example: push

Push a 3:
First example: push

StackExample>>shouldPush

| stack |

stack := Stack new.

stack push: 3
First example: push

Push a 4:
First example: push

StackExample>>shouldPush
  | Stack |
stack := Stack new.
stack push: 3; push: 4
First example: push

Now:

stack size should = 2
First example: push

StackExample>>shouldPush
| stack |
stack := Stack new.
stack push: 3; push: 4.
stack size should = 2.
StackExample>>shouldPush
| stack |
stack := Stack new.
stack push: 3; push: 4.
stack size should = 2.
^ stack

Make it expandable!
Second example: pop

Start with the previous stack:
Second example: pop

StackExample>>shouldPop

| stack |

stack :=

self given: #shouldPush
Second example: pop

Pop the 4:
Second example: pop

StackExample>>shouldPop

| stack |
stack :=
self given: #shouldPush.
stack pop.
Second example: pop

Now:

stack size should = 1
Second example: pop

StackExample>>shouldPop
| stack |
stack :=
    self given: #shouldPush.
stack pop.
stack size should = 1
Second example: pop

StackExample>>shouldPop
  | stack |
  stack :=
    self given: #shouldPush.
  stack pop.
  stack size should = 1.
  ^ stack
What would you do in JUnit?
push test would be about the same
but **POP** would differ!
void testPop() {
    Stack stack = new Stack();
    stack.push(3); stack.push(4);
    stack.pop();
    assertEquals(stack.size(), 1);
}
void testPop() {
    Stack stack = new Stack();
    stack.push(3); stack.push(4);
    stack.pop();
    assertEquals(stack.size(), 1);
}

Weren’t we testing pop?
Can only pop if there is something
Won brevity
shouldPush \quad \text{expands on} \quad \text{shouldPop}

execute only if shouldPush works
Effect:
JUnit

Push broken

Push test fails

Pop broken

Pop test fails

Mittwoch, 9. Dezember 2009
Phexample

Push broken

Push test fails

Pop broken

Pop test fails
improved locality
Summary
good syntax

(almost like human language!)
improves brevity
(reduce tests to the interesting part)
good locality

(One bug leads to one failing examples!)
KTHXBYE

questions?
These slides are available under the GFDL.