



Jiří KNEŠL



COFFEE SCRIPT

PROC?

JAK?



```
a = "uvnitr retezce #{5 + 5}"  
a.replace /(10)/, "11"
```

```
if x is 55  
    objekt.metoda()
```

```
metoda = (par1, par2) ->  
    par1.x() + par2.y()
```

```
class Def extends Abc
```

```
    constructor: ->  
        @abc = 11
```

```
    metoda: -> @abc + 5
```

```
a = "uvnitr retezce " + (5 + 5);  
a.replace /(10)/, "11";
```

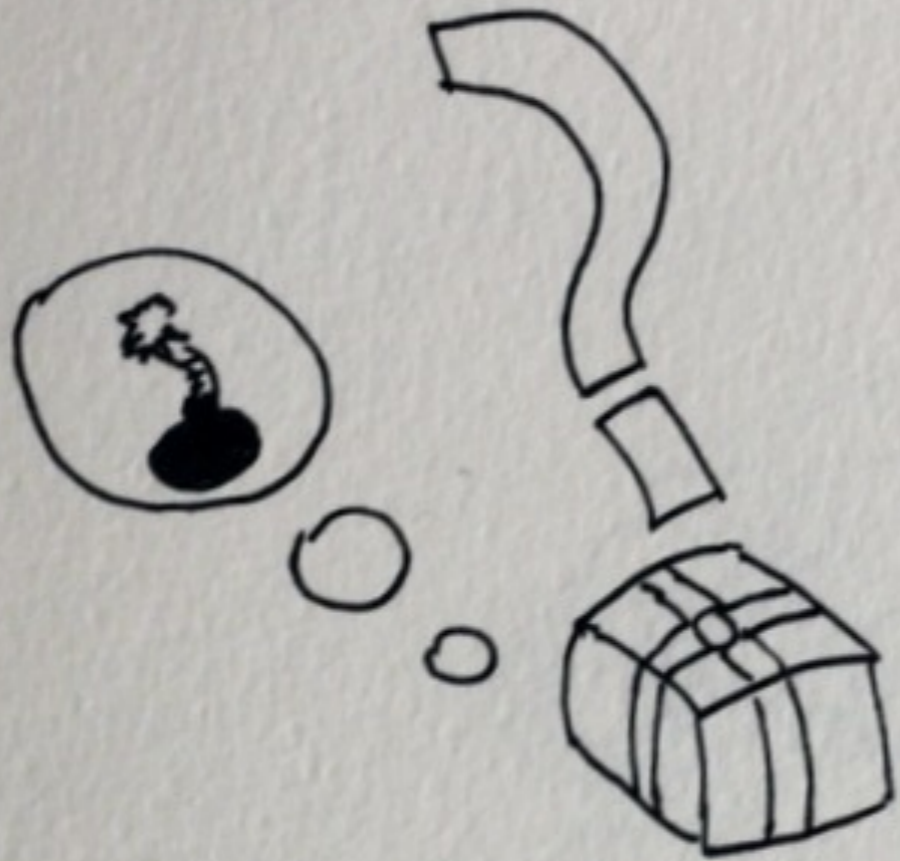
```
if (x === 55) {  
    objekt.metoda();  
}
```

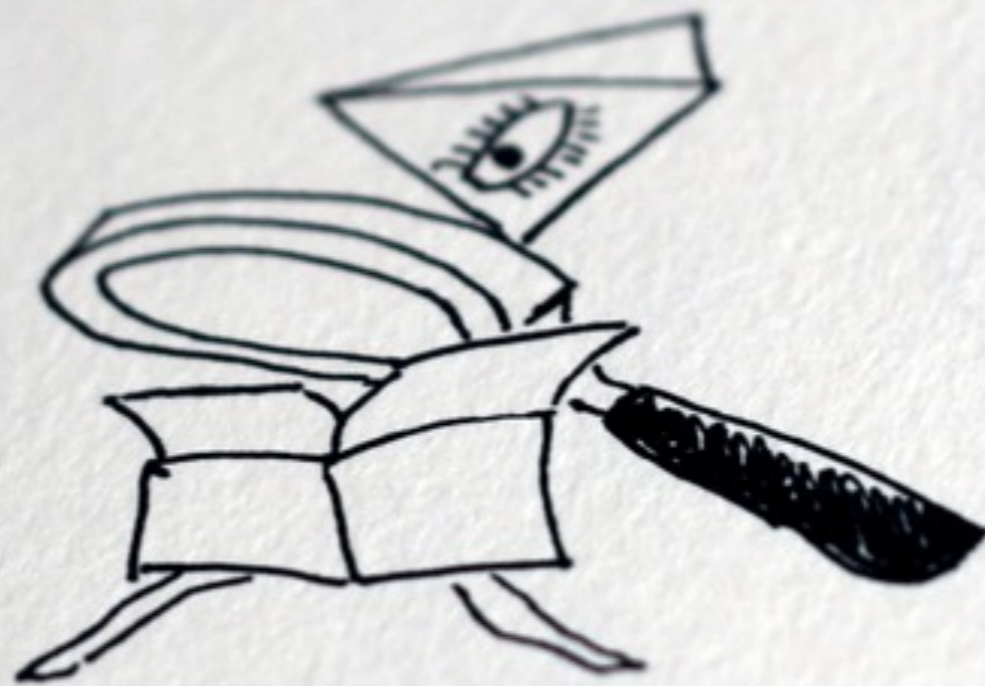
```
metoda = function(par1, par2) {  
    return par1.x() + par2.y();  
};
```

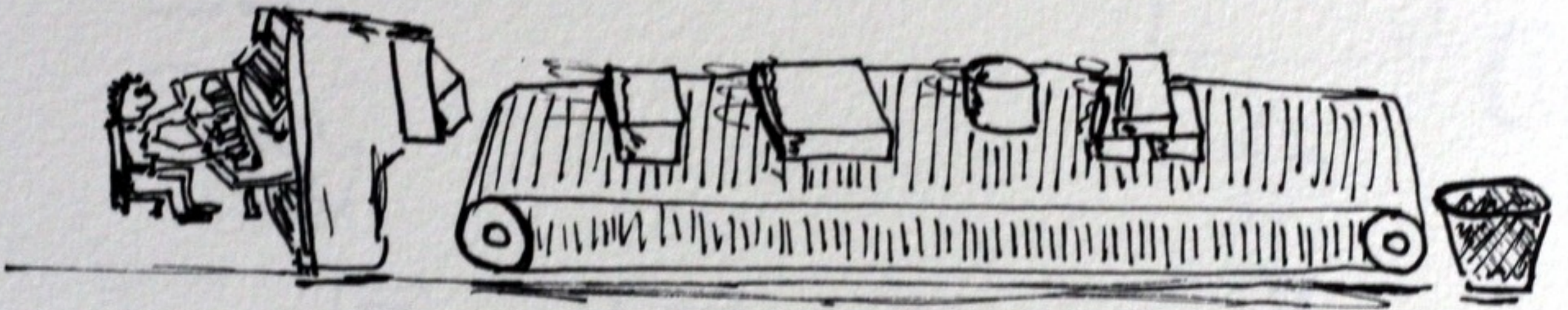
```
Def = (function() {  
    __extends(Def, Abc);  
    function Def() {  
        this.abc = 11;  
    }  
})
```

```
Def.prototype.metoda = function() {  
    return this.abc + 5;  
};  
return Def;  
})();
```

Proč?







Jak na to?

```
test "a == 1" ->  
    equal a, 1
```

```
test "document.ready", ->
  $ ->
    a = 2
    start()
    $(document).trigger "ready"
    equal a, 2
```

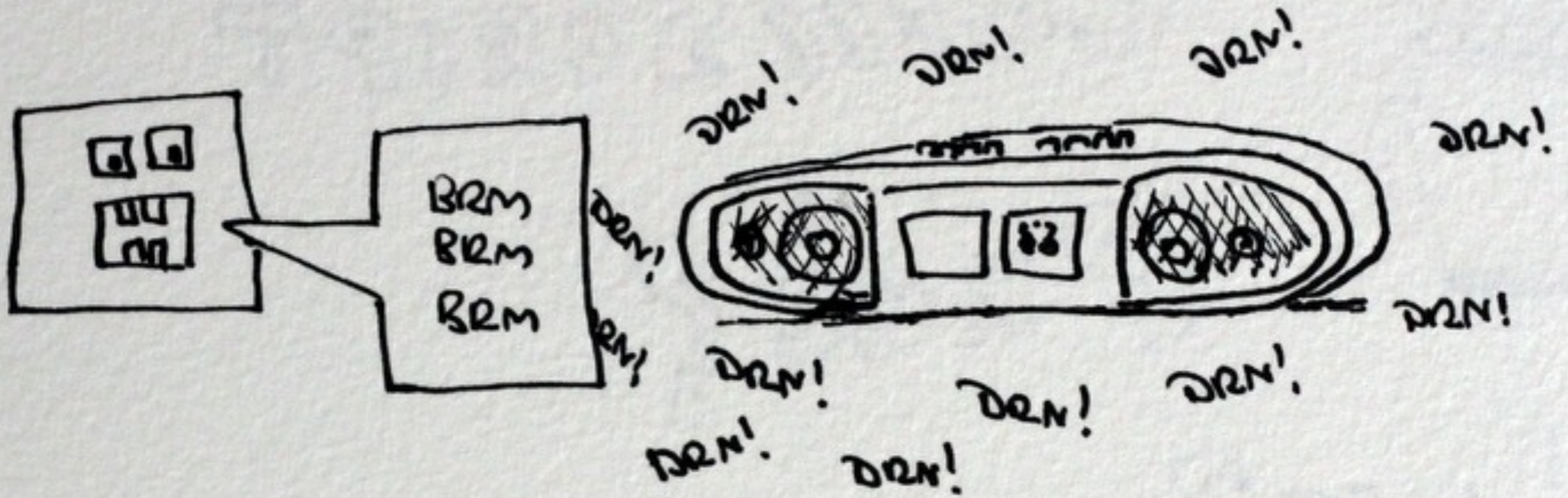
```
asyncTest "onclick", ->
  $("body").click ->
    a = 3
    equal a, 3
    start()
  $("body").trigger "click"
```

```
asyncTest "ajax post", ->  
  $.post "http://localhost/jiriknesl/webexpo11/ajax.php", {a: "x"}, (result) ->  
    equal result, JSON.stringify {a: "x"}  
  start()
```

```
asyncTest "Test my ajax", ->
  $.extend "post": -> {abc: 11}

  $.post "myurl", {}, (result) ->
    equals 11, result.abc
```





```
test "sinonMock", ->
  myApi = method: (a) -> a*2
  mock = @mock myApi
  mock.expects("method").once().returns(5)
  ret = myApi.method(2)
  equal 5, ret
```

```
test "mock test", ->
```

```
  class Tested
```

```
    createX: -> new X
```

```
    sdf: ->
```

```
      z = @createX()
```

```
      z.d() + 1
```

```
  tested = new Tested
```

```
  tested.createX = =>
```

```
    out = d: -> 12
```

```
    mock = @mock out
```

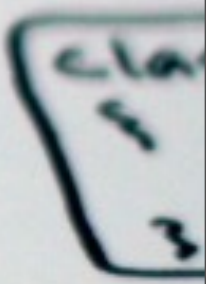
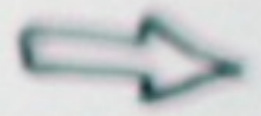
```
    mock.expects("d").once().returns(12)
```

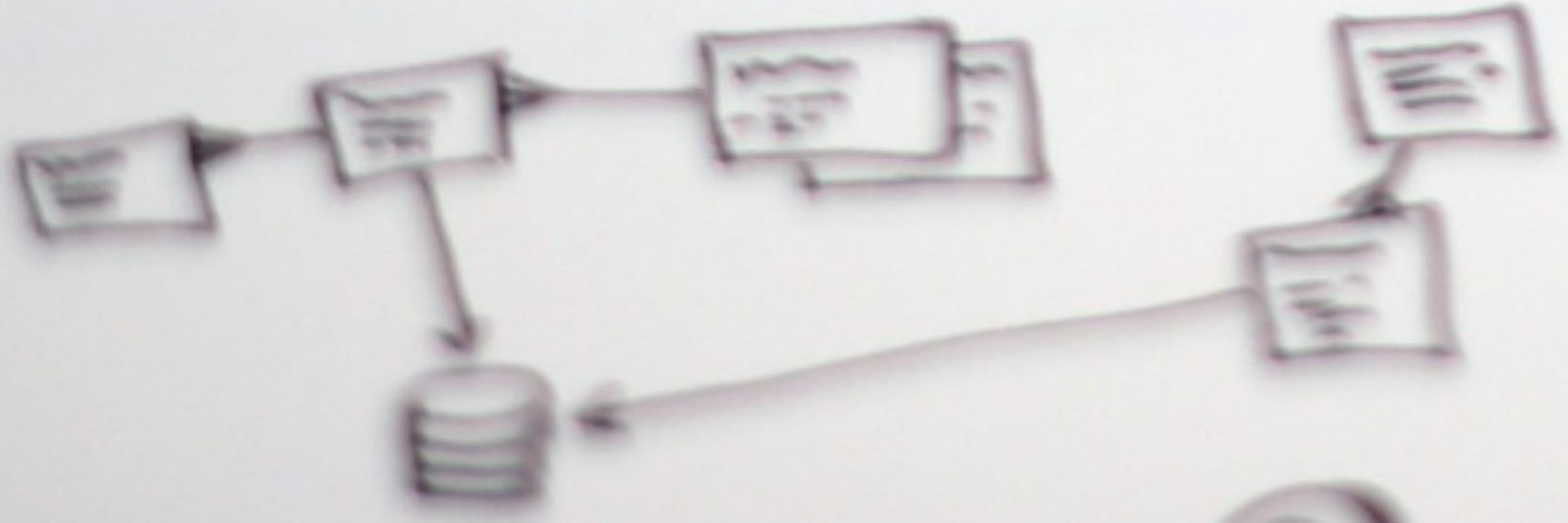
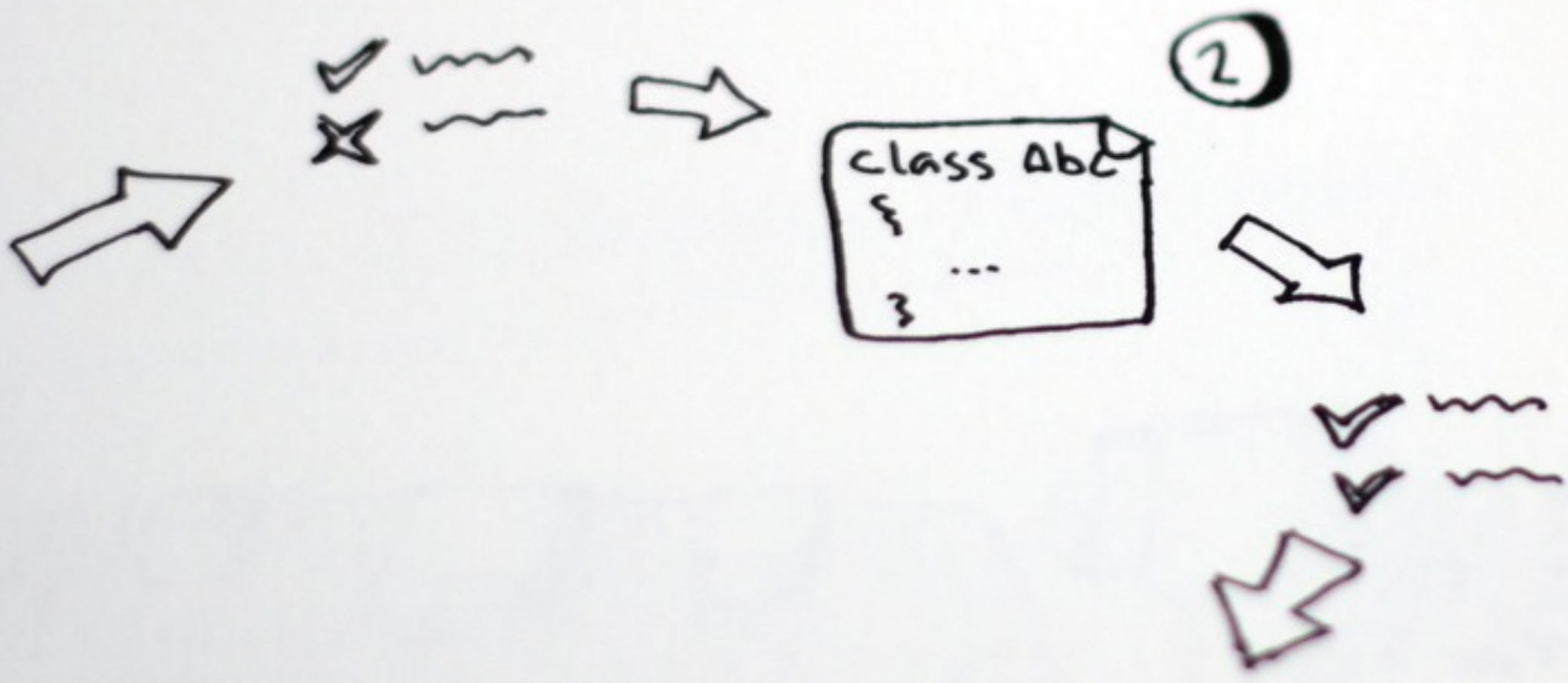
```
    out
```

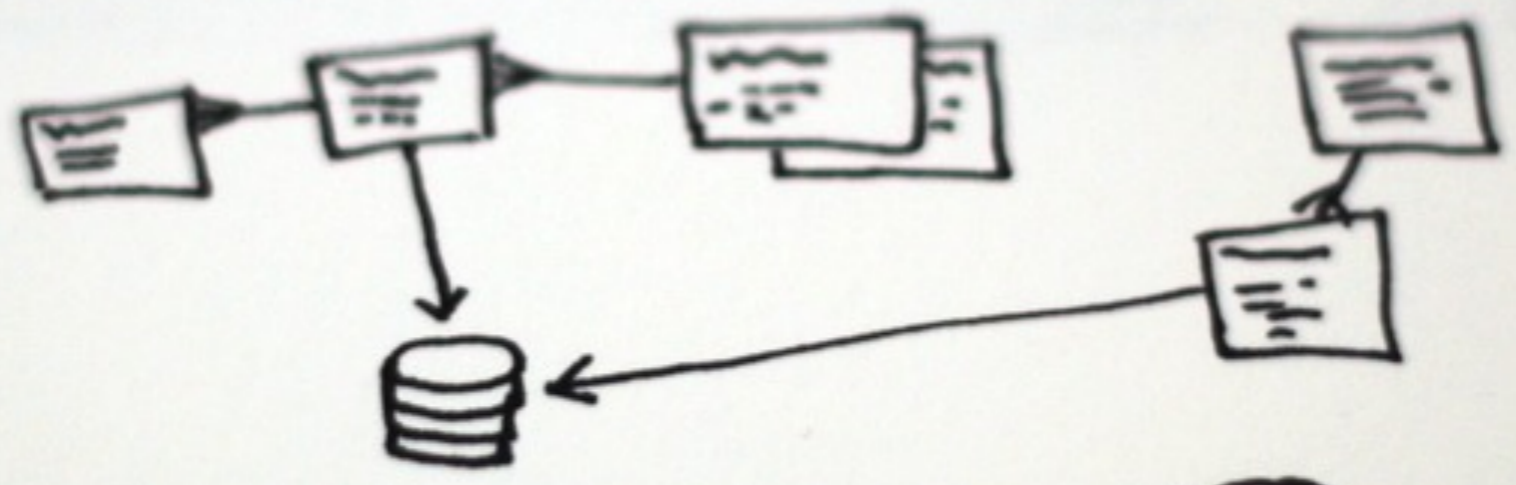
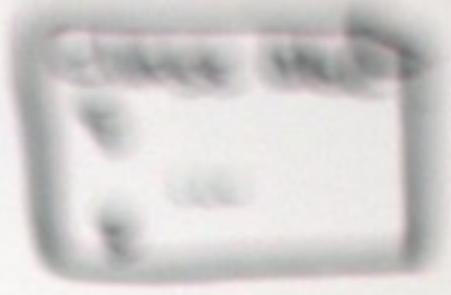
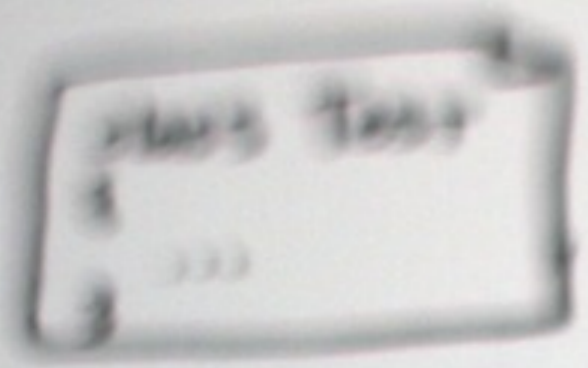
```
  equals tested.sdf(), 13
```

①

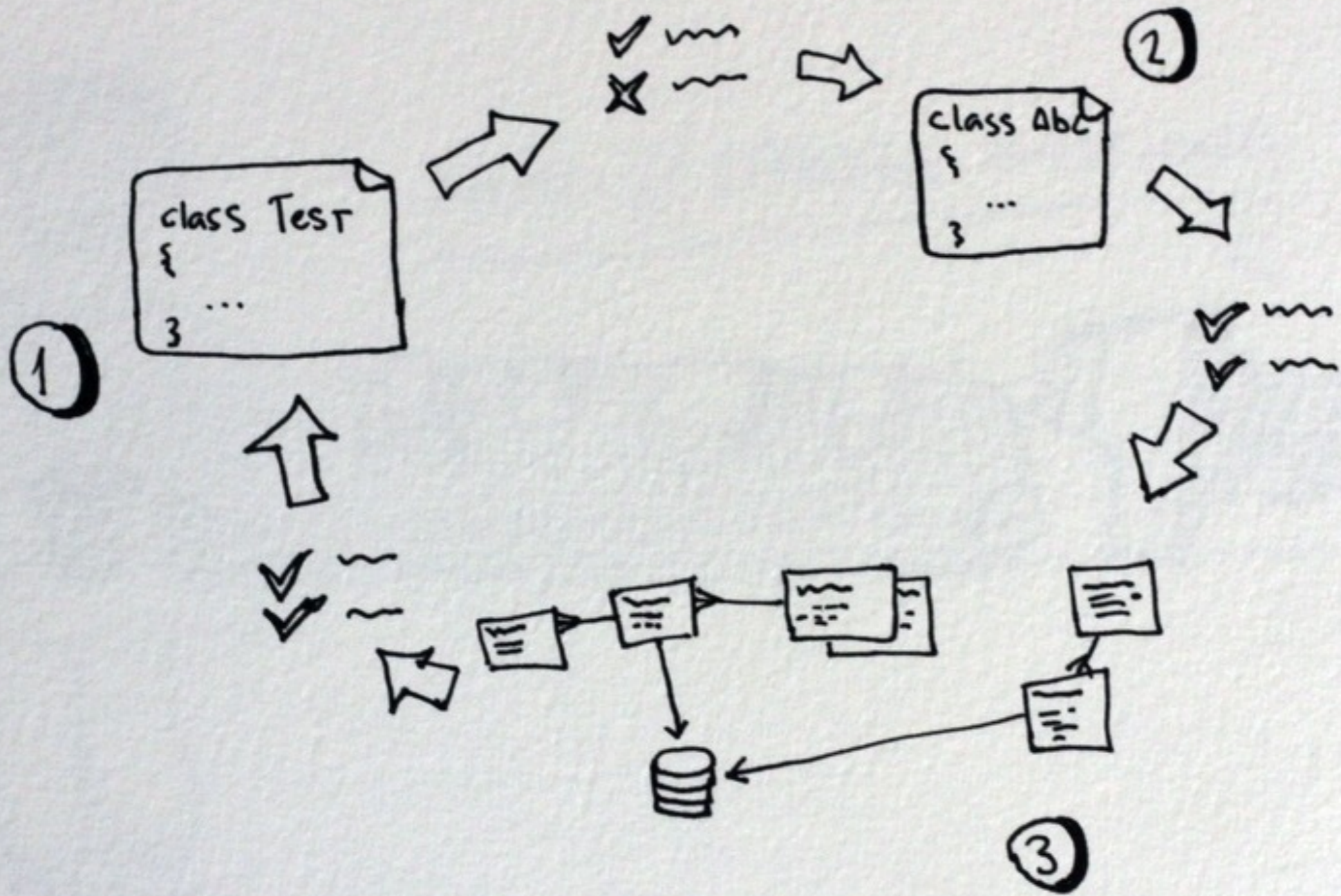
```
class Test  
{  
  ...  
}
```







3



# Testovatelný kód



<http://ukazka.knesl.com>