



Разработка через тестирование Approval Tests

Test Driven Development

Ivan Dyachenko <IDyachenko@luxoft.com>

Содержание

1

Approval Testing Library

2

Legacy code

3

Шаблон Approval теста

4

Пример разработки через Approval TDD

5

Reporters

6

Где это работает

7

Workshop на тему “Locking down legacy code”

Approval Tests



Approval Tests как альтернативный взгляд на
автоматизированное тестирование

Approval Tests



Llewellyn Falco

- Professional Speaker
- Teacher & Agile Programmer
- Agile Coach
- Creator of ApprovalTest
- Co-Founder of TeachingKids Programming
- Legacy Code Expert

<http://approvaltests.com>

<http://llewellynfalco.blogspot.com>

Approval Tests



Approval Testing Library предлагает комбинировать
сильные свойства компьютеров и человека, для
достижения высокого качества тестирования

Legacy code



Legacy code is the code that...

Works!

Legacy code



Legacy code has no tests...

Шаблон теста

Test...

```
{  
    // Arrange  
    ...  
    // Action  
    ...  
    // Assertion  
    ...  
}
```

Should...

```
{  
    // Given  
    ...  
    // When  
    ...  
    // Then  
    ...  
}
```

Approval...

```
{  
    // Do  
    ...  
    // Verify  
}
```


// **DO** by computer

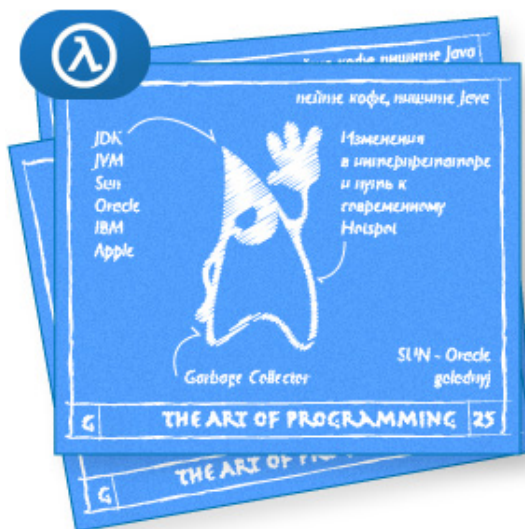
// **VERIFY** by human

Legacy code



Approval Tests: Locking Down Legacy Code

Пишем код



Legacy code

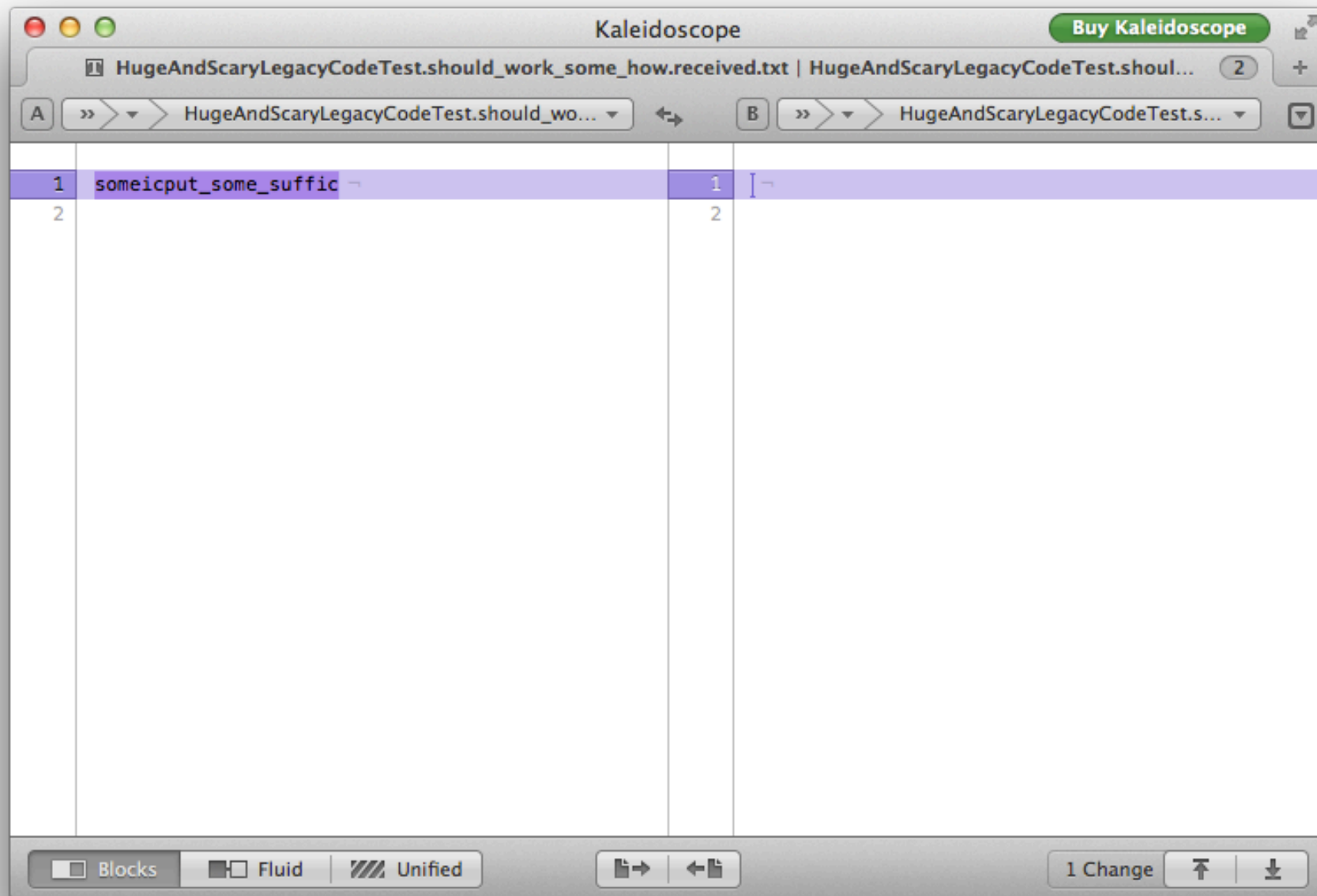
```
public static String TheUgliesMethodYouMightEverSeen(String s, int i, char c) {  
    if (s.length() > 5) {  
        s += "_some_suffix";  
    }  
  
    StringBuilder r = new StringBuilder();  
    for (int j = 0; j < s.length(); j++) {  
        char k = s.charAt(j);  
        if ((int) k % i == 0) {  
            r.append(c);  
        } else {  
            if (k == c) {  
                if (r.length() <= 2) {  
                    r.append('a');  
                } else {  
                    r.append('b');  
                }  
            }  
            if (k == '^') {  
                r.append('c');  
            } else {  
                r.append(k);  
            }  
        }  
    }  
  
    return r.toString();  
}
```

Первый простой тест

```
@Test
public void should_work_some_how() throws Exception {
    verify(
        HugeAndScaryLegacyCode.
        TheUgliesMethodYouMightEverSeen("someinput", 10, 'c')
    );
}
```

Все, что нам надо –
вызвать метод
Approvals.verify

Результат выполнения тестов



Coverage

```
public static String TheUgliesMethodYouMightEverSeen(String s, int i, char c) {  
    if (s.length() > 5) {  
        s += "_some_suffix";  
    }  
  
    StringBuilder r = new StringBuilder();  
    for (int j = 0; j < s.length(); j++) {  
        char k = s.charAt(j);  
        if ((int) k % i == 0) {  
            r.append(c);  
        } else {  
            if (k == c) {  
                if (r.length() <= 2) {  
                    r.append('a');  
                } else {  
                    r.append('b');  
                }  
            }  
            if (k == '^') {  
                r.append('c');  
            } else {  
                r.append(k);  
            }  
        }  
    }  
  
    return r.toString();  
}
```

У нас все еще есть
непокрытый тестами
код

Use combinations of arguments

```
@Test
public void should_try_to_cover_it() throws Exception {
    int[] numbers = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    char[] chars =
        "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ".toCharArray();
    String[] strings = {"", "approvals", "xpdays", "^stangeword^"};
    List<String> result = new ArrayList<String>();

    for (int number : numbers) {
        for (char c : chars) {
            for (String string : strings) {
                result.add(
                    HugeAndScaryLegacyCode.
                        TheUgliesMethodYouMightEverSeen(string, number, c)
                );
            }
        }
    }

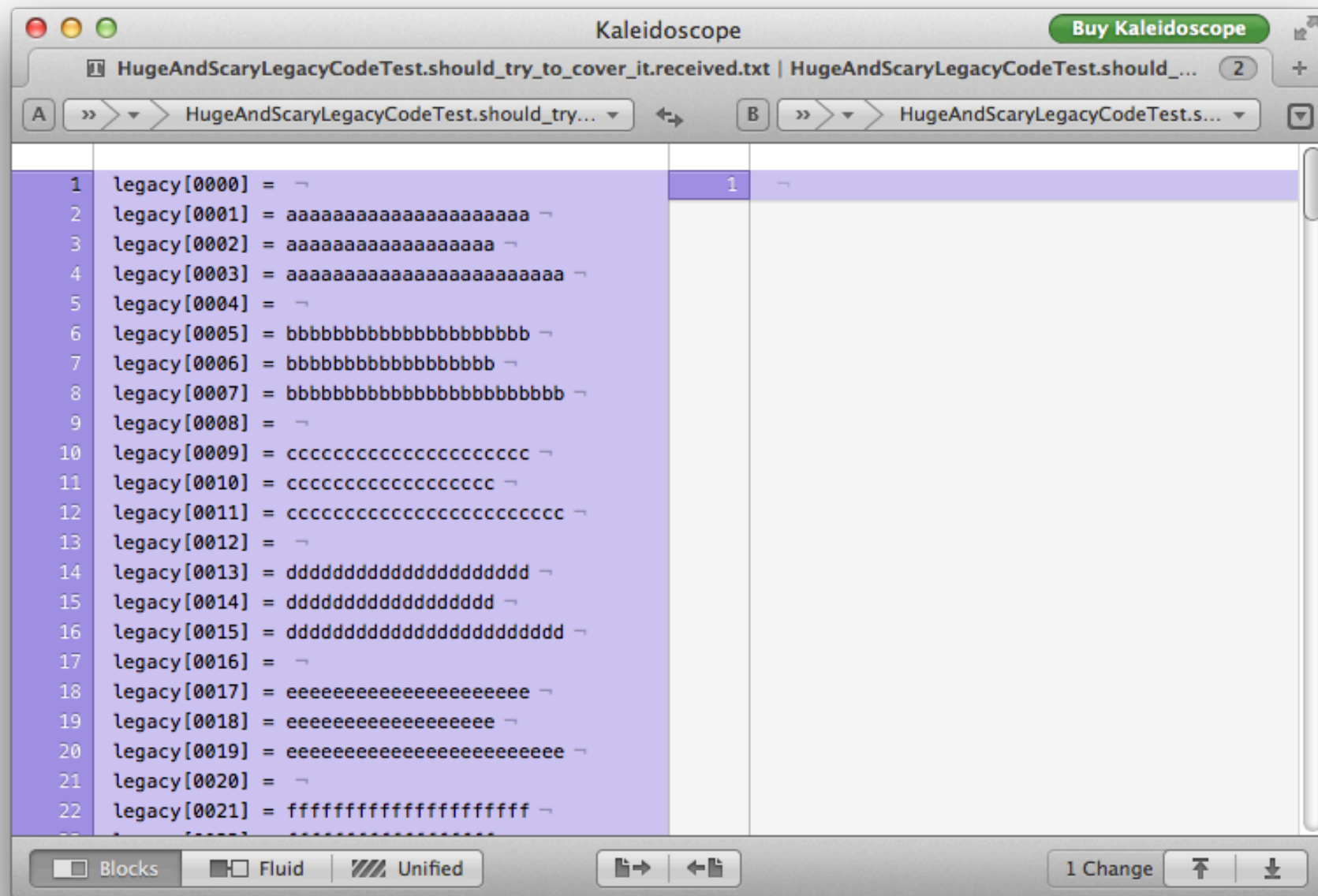
    verifyAll("legacy", result.toArray());
}
```


Coverage

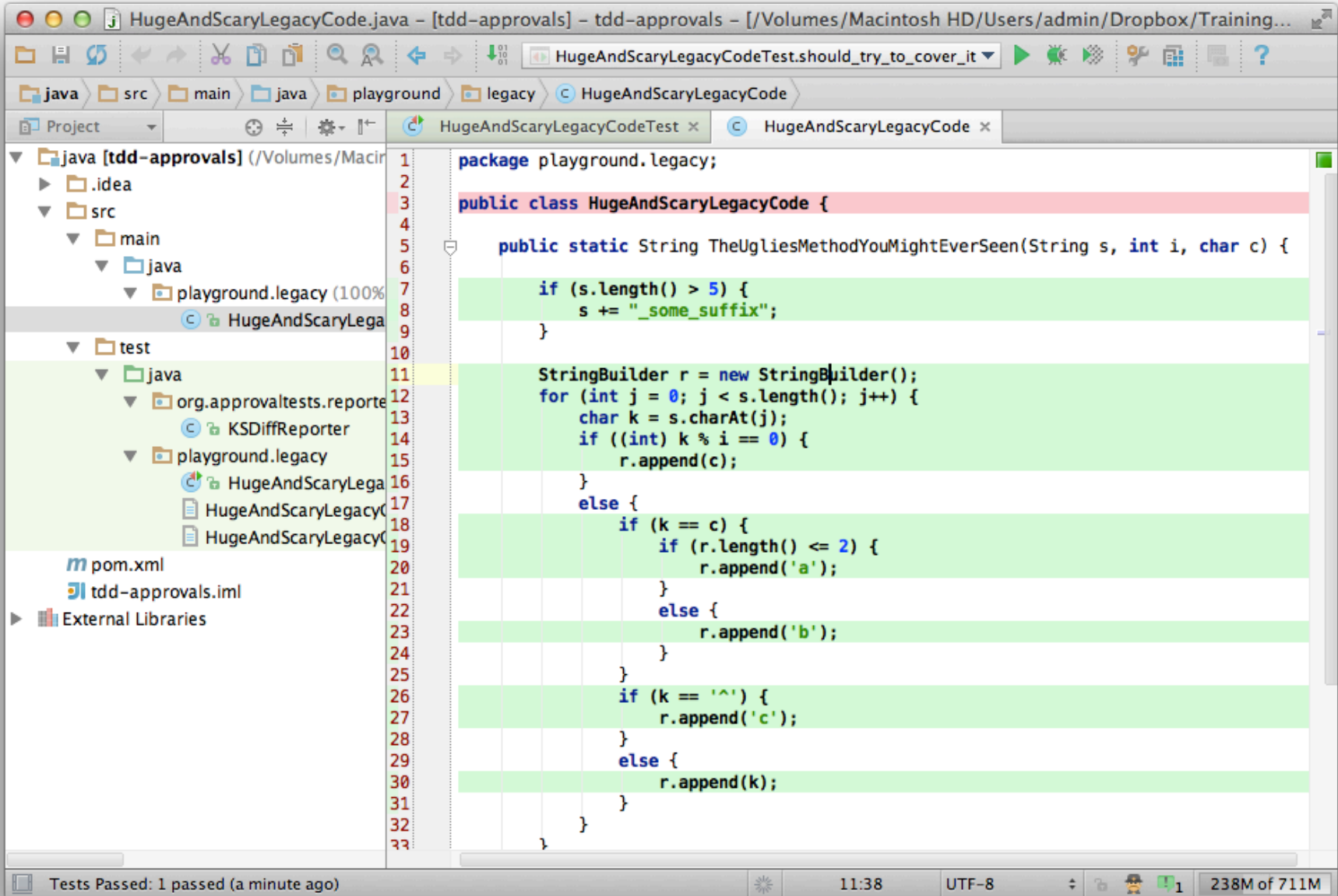


Написав пару строк кода мы получили 2079
тестов и все они корректные!

Результат



Coverage 100%



The screenshot shows an IDE window titled "HugeAndScaryLegacyCode.java - [tdd-approvals] - tdd-approvals - [/Volumes/Macintosh HD/Users/admin/Dropbox/Training...". The left sidebar displays a project tree for "java [tdd-approvals] (/Volumes/Macintosh HD/Users/admin/Dropbox/Training...". The tree includes folders for ".idea", "src", and "test". Under "src", there is a "main" folder containing a "java" folder, which in turn contains a "playground.legacy" folder (100% coverage) and a "HugeAndScaryLegacyCode" class. Under "test", there is a "java" folder containing "org.approvaltests.reporter" (with "KSDiffReporter" class) and "playground.legacy" (with "HugeAndScaryLegacyCode" class). The main editor shows the source code of "HugeAndScaryLegacyCode.java". The code is as follows:

```
1 package playground.legacy;
2
3 public class HugeAndScaryLegacyCode {
4
5     public static String TheUgliesMethodYouMightEverSeen(String s, int i, char c) {
6
7         if (s.length() > 5) {
8             s += "_some_suffix";
9         }
10
11         StringBuilder r = new StringBuilder();
12         for (int j = 0; j < s.length(); j++) {
13             char k = s.charAt(j);
14             if ((int) k % i == 0) {
15                 r.append(c);
16             }
17             else {
18                 if (k == c) {
19                     if (r.length() <= 2) {
20                         r.append('a');
21                     }
22                     else {
23                         r.append('b');
24                     }
25                 }
26                 if (k == '^') {
27                     r.append('c');
28                 }
29                 else {
30                     r.append(k);
31                 }
32             }
33         }
34     }
35 }
```

The status bar at the bottom indicates "Tests Passed: 1 passed (a minute ago)", "11:38", "UTF-8", and "238M of 711M".

Locking down



Процесс контролирования legacy кода
подобным образом называется
"Locking down".

Shopping Cart Tests

Приведем другой пример:

К примеру у нас есть class **ShoppingCart** и мы можем добавлять различные продукты в корзину.

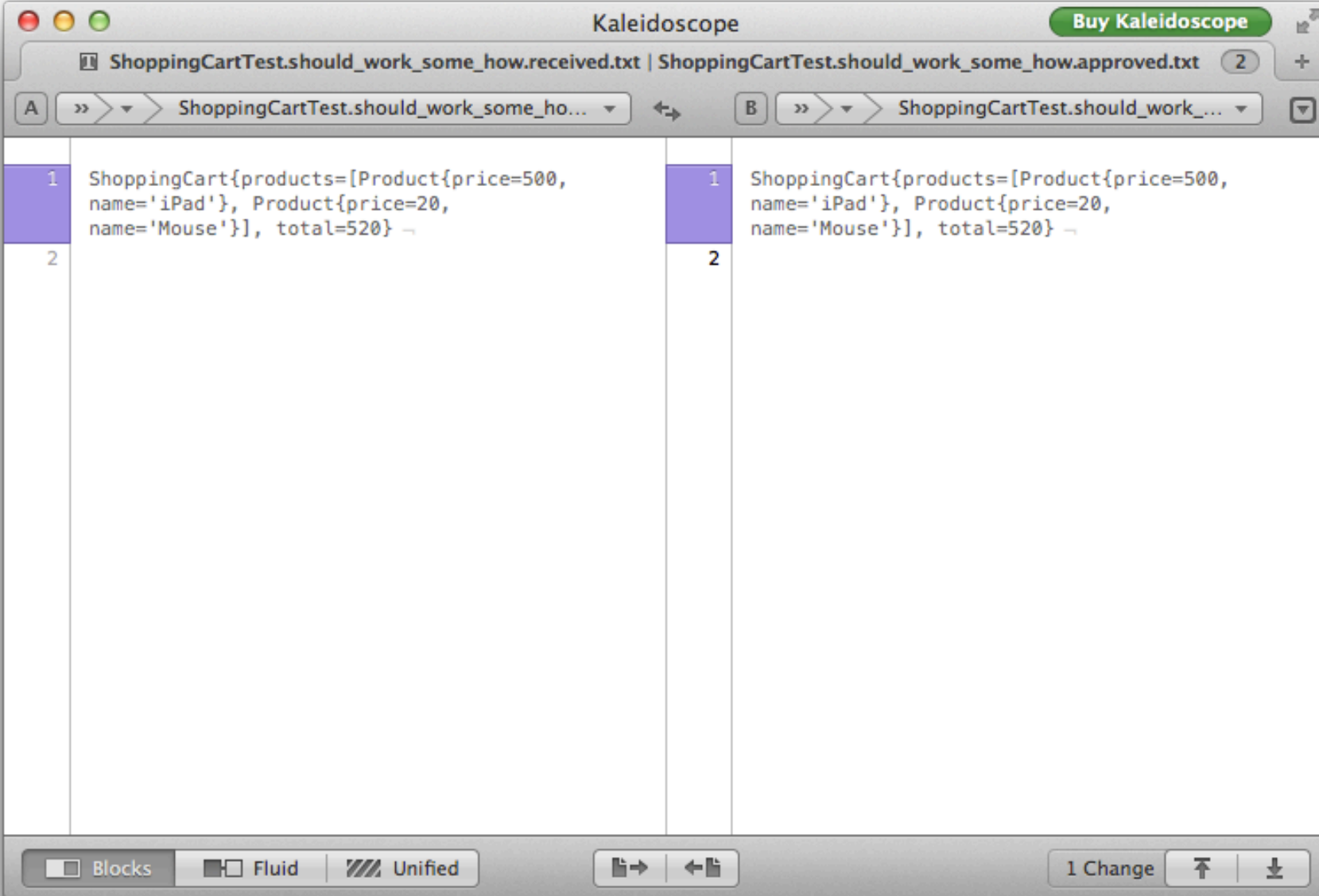
Надо протестировать, что **confirm** считается верно.

Shopping Cart Tests

@Test

```
public void should_work_some_how() throws Exception {  
    // do  
    ShoppingCart shoppingCart = new ShoppingCart();  
    shoppingCart.add(new Product("iPad", 500));  
    shoppingCart.add(new Product("Mouse", 20));  
    shoppingCart.confirm();  
  
    // verify  
    verify(shoppingCart.toString());  
}
```

Результат



Kaleidoscope

Buy Kaleidoscope

ShoppingCartTest.should_work_some_how.received.txt | ShoppingCartTest.should_work_some_how.approved.txt 2

A >> ShoppingCartTest.should_work_some_ho... <-> B >> ShoppingCartTest.should_work_... <v>

1	ShoppingCart{products=[Product{price=500, name='iPad'}, Product{price=20, name='Mouse'}], total=520} -	1	ShoppingCart{products=[Product{price=500, name='iPad'}, Product{price=20, name='Mouse'}], total=520} -
2		2	

Blocks Fluid Unified

1 Change

Reporters

@UseReporter(Reporter.**class**)

DiffReporter	Launches an instance of TortoiseSvnDiff
FileLauncherReporter	Opens the .received file
ImageDiffReporter	Launches an instance of TortoiseSvnImageDiff
ImageWebReporter	Opens the files in a web browser
JunitReporter	Text only, displays the contents of the files as a AssertEquals failure
NotePadLancher	Opens the .received file in notepad
QuietReporter	Outputs the move command to the console. Great for build systems
TextWebReporter	Opens the files in a web browser

When it works?



Лучше всего это работает для:

- UI
- Legacy Code

When it works?

Везде где вы можете представить результат работы кода в виде:

- HTML
- XML
- JSON
- SQL
- Other output

Workshop





Вопросы ?



Разработка через тестирование

IDyachenko@luxoft.com

```
git clone git://github.com/ivan-dyachenko/Trainings.git
```

```
https://github.com/ivan-dyachenko/Trainings
```