



Урок Спринт 3.1

Флаг -count

Команда `go test` позволяет запускать тесты несколько раз. Давайте попробуем запустить тесты 3 раза:

```
go test -v -count=3
```

Зачем это нужно? Есть случаи, когда тест падает не в 100% запусков. Причиной этого может быть то, что тест зависит от внешних факторов, например, от базы данных. Если тест проходит успешно не в 100% запусков и при отсутствии внешних факторов, то это значит что тест написан неверно.

Давайте изменим код программы, чтобы сохранять имена в типе `map`:

```
package printer

import "fmt"

var names = make(map[string]string)

func PrintHello(name string) string {
    names[name] = name
    return fmt.Sprintf("Hello, %s!", name)
}
```

Запустим тесты 3 раза:

```
go test -v -count=3
```

Видим, что тесты проходят успешно:

```
ivan.petrov ~/project $ go test -v -count=3
=== RUN   TestPrintHello
--- PASS: TestPrintHello (0.00s)
=== RUN   TestPrintHello
--- PASS: TestPrintHello (0.00s)
=== RUN   TestPrintHello
--- PASS: TestPrintHello (0.00s)
PASS
ok      module3_1-lesson      0.096s
```

Теперь давайте изменим тест так, чтобы он проверял что в `map` есть имя `Igor`:

```
func TestPrintHello(t *testing.T) {
    got := PrintHello("Igor")
    expected := "Hello, Igor!"

    if got != expected {
        t.Fatalf(`PrintHello("Igor") = %q, want %q`, got, expected)
    }

    for name := range names {
        if name != "Igor" {
            t.Fatalf(`got %q, want %q`, name, "Igor")
        } else {
            break
        }
    }
}
```

Тесты опять проходят успешно:

```
ivan.petrov ~/project $ go test -v -count=3
=== RUN    TestPrintHello
--- PASS: TestPrintHello (0.00s)
=== RUN    TestPrintHello
--- PASS: TestPrintHello (0.00s)
=== RUN    TestPrintHello
--- PASS: TestPrintHello (0.00s)
PASS
ok      module3_1-lesson      0.096s
```

Однако, если мы добавим еще одно имя в map names:

```
func TestPrintHello(t *testing.T) {
    names["Petr"] = "Petr"
    got := PrintHello("Igor")
    expected := "Hello, Igor!"

    if got != expected {
        t.Fatalf(`PrintHello("Igor") = %q, want %q`, got, expected)
    }

    for name := range names {
        if name != "Igor" {
            t.Fatalf(`got %q, want %q`, name, "Igor")
        } else {
            break
        }
    }
}
```

Тест упадёт:

```
=== RUN    TestPrintHello
```

```
main_test.go:18: got "Petr", want "Igor"
--- FAIL: TestPrintHello (0.00s)
=== RUN TestPrintHello
main_test.go:18: got "Petr", want "Igor"
--- FAIL: TestPrintHello (0.00s)
=== RUN TestPrintHello
--- PASS: TestPrintHello (0.00s)
FAIL
exit status 1
FAIL    module3_1-lesson    0.089s
```

Запустите тест несколько раз и убедитесь, что тест падает не с первого раза в ряде случаев:

```
go test -v -count=3
```

Иногда это будет выглядеть так:

```
ivan.petrov ~/project $ go test -v -count=3
=== RUN TestPrintHello
--- PASS: TestPrintHello (0.00s)
=== RUN TestPrintHello
main_test.go:18: got "Petr", want "Igor"
--- FAIL: TestPrintHello (0.00s)
=== RUN TestPrintHello
--- PASS: TestPrintHello (0.00s)
FAIL
exit status 1
FAIL    module3_1-lesson    0.089s
```

Или даже так:

```
ivan.petrov ~/project $ go test -v -count=3
=== RUN TestPrintHello
--- PASS: TestPrintHello (0.00s)
=== RUN TestPrintHello
--- PASS: TestPrintHello (0.00s)
=== RUN TestPrintHello
--- PASS: TestPrintHello (0.00s)
FAIL
ok      module3_1-lesson    2.285s
```

Поясните, почему тест падает и как это исправить?

`PrintHello` добавляет имя в `map names`. Встроенный оператор `range` возвращает элементы в случайном порядке. При каждом запуске теста порядок элементов в `map names` будет разным. Поэтому тест падает. Несмотря на простоту примера, эта ошибка часто встречается в реальных проектах, даже когда код пишут опытные разработчики, будьте внимательны.

По завершению упражнения откатите изменения в тесте `TestPrintHello`.

Справка

Исключительное право на учебную программу и все сопутствующие ей учебные материалы, доступные в рамках сервиса, принадлежат АНО ДПО «Образовательные технологии Яндекса». Воспроизведение, копирование, распространение и иное использование программы и материалов допустимо только с предварительного письменного согласия АНО ДПО «Образовательные технологии Яндекса».

[Пользовательское соглашение.](#)

© 2018 – 2024 ООО «Яндекс»