Unit test-ове в Haskell

доц. Атанас Семерджиев

1

HUnit

Инсталация (вариант 1)

```
cabal update
cabal install HUnit
```

Инсталация (вариант 2)

1. Пакетът се сваля от https://hackage.haskell.org/package/HUnit
2. Сваля се и `Call-Stack`: https://hackage.haskell.org/package/call-stack
3. Пакетите се разархивират и в директорията на всеки се изпълнява:

```
runhaskell Setup.hs configure
runhaskell Setup.hs build
runhaskell Setup.hs install
```

2

2

Test Cases

```
import Test.HUnit

fact 0 = 1
fact n = n * fact (n - 1)

test1 = TestCase (assertEqual "0! = 1"   1   (fact 0))
test2 = TestCase (assertEqual "1! = 1"   1   (fact 1))
test3 = TestCase (assertEqual "5! = 120" 120 (fact 5))
```

3

3

Test Cases

```
import Test.HUnit

fact 0 = 1
fact n = n * fact (n - 1)

test1 = TestCase $ assertEqual "0! = 1"    1   (fact 0)
test2 = TestCase $ assertEqual "1! = 1"    1   (fact 1)
test3 = TestCase $ assertEqual "5! = 120"  120 (fact 5)
```

4

4

Test Cases

```
test4 = TestCase $ do
    assertBool "true is true" True
    assertEqual "1 = 1" 1 1
    assertFailure "This will surely fail!"
```

```
test5 = TestCase $ do
    1 @=? (fact 1)
    (fact 0) @?= 1
    True @? "True is true"
    False @? "False is true"
```

5

5

Съкратен запис

```
test6 = 1 ~=? (fact 1)
test7 = (fact 1) ~=? 1
test8 = False ~? "False is true"
```

```
test6 = TestLabel "1 = 1!" $ 1 ~=? (fact 1)
test7 = TestLabel "1! = 1" $ (fact 1) ~?= 1
test8 = TestLabel "Fail"   $ False ~? "False is true"
```

```
test6 = "1 = 1!" ~: 1 ~=? (fact 1)
test7 = "1! = 1" ~: (fact 1) ~?= 1
test8 = "Fail"   ~: False ~? "False is true"
```

6

6

```
TestLabel label t

TestLabel "msg“

"1 = 1!" ~: 1 ~=? (fact 1)
```

7

7

Пускане на тестовете

```
tl = TestList [test1,test2,test3,test4,test5,
               test6,test7,test8,test9]

main = runTestTT tl
```

8

8

Пускане на тестовете

```
tl = TestList [test1,test2,test3,test4,test5,
               test6,test7,test8,test9]
```

```
main = runTestTT tl
```

9

9

```
tl2 = TestList [

    "0! = 1" ~:
        (fact 0) ~?= 1,

    "1! = 1" ~:
        (fact 1) ~?= 1,


    "5! = 120" ~:
        (fact 5) ~?= 120
    ]
```

```
main = do
    runTestTT tl
    runTestTT tl2
```

10

10

11