TEST, TRANSFORM, REFACTOR

PHPUK 2014

by @_md

Learning

@_md

Marcello Duarte

Lead developer

Head of Training

$$\text{Kata}\left\{\begin{array}{l}\text{form}\\ \text{movement}\\ \text{practice}\\ \text{deliberate}\end{array}\right.$$

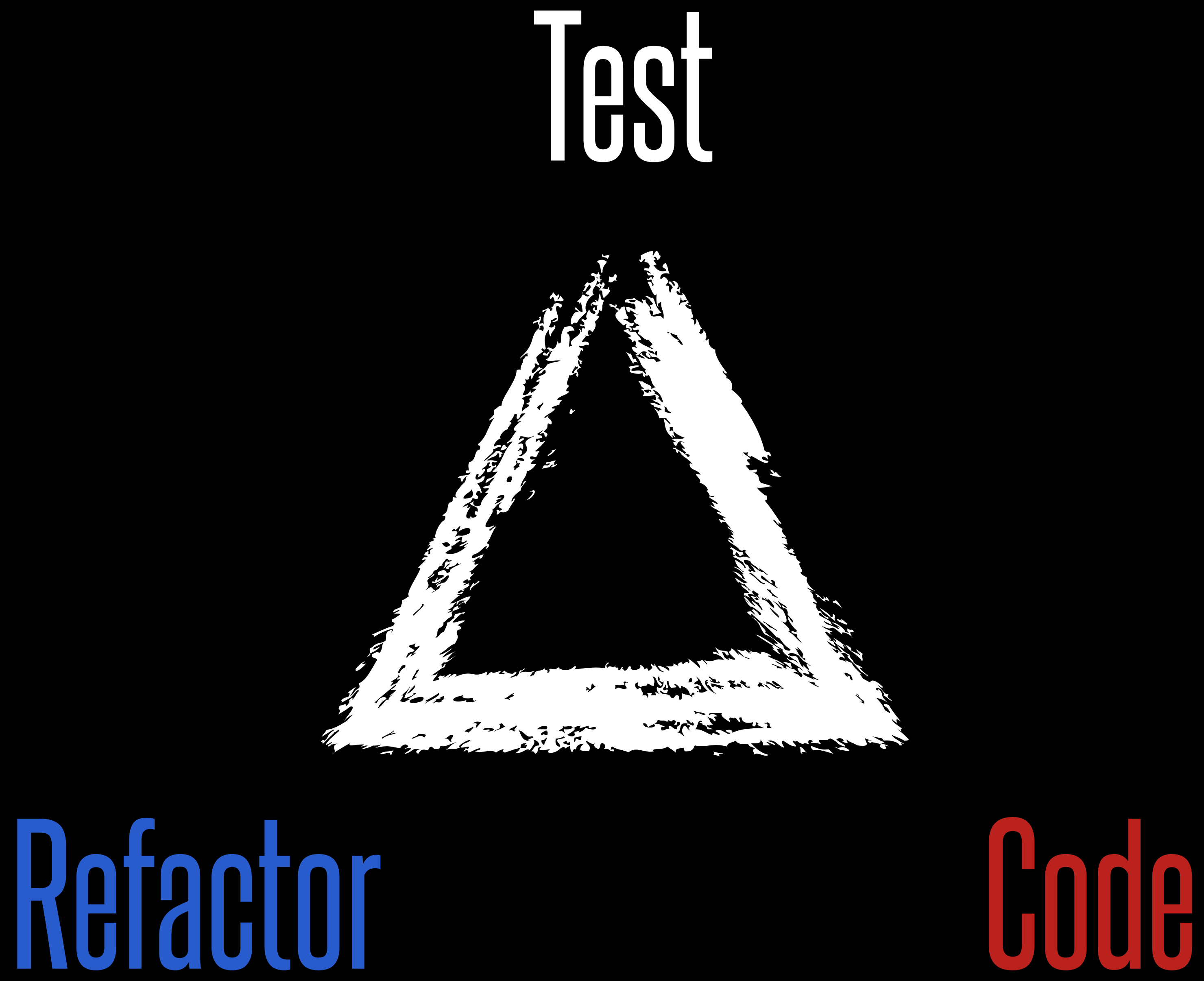
20 minutes

throw away

TDD

refactor

$\Bigg\}$ Kata


Test
Refactor
Code

TDD 3Rules

- no code without test
- enough to go red
- enough to go green

Test

Code

no code without test

enough to go red

enough to go green

Refactor

Simple Design

4Rules

- tests run and pass
- dry
- remove opacity
- simplify

The Roman Numerals Kata

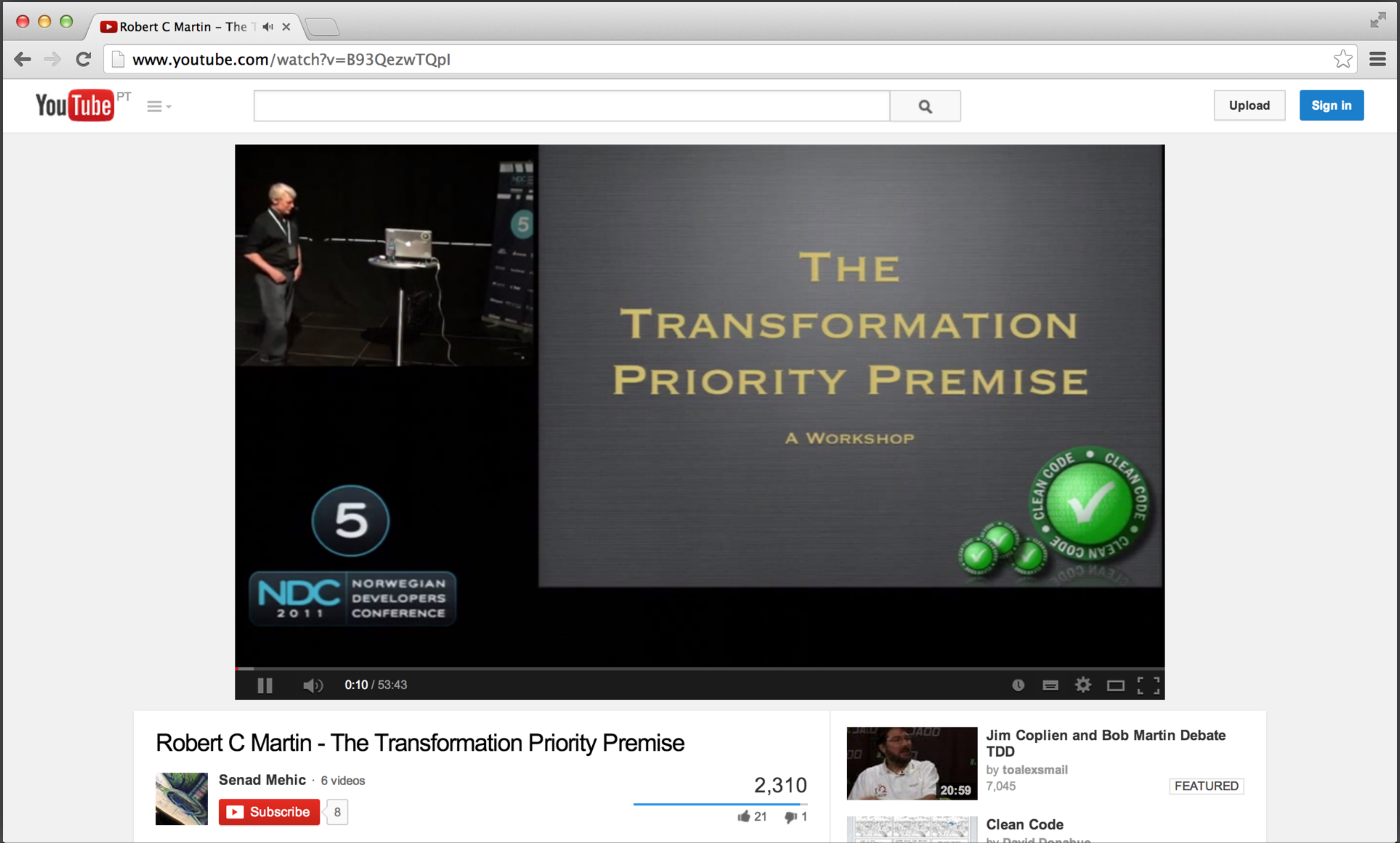
Converting arabic numbers into their roman equivalent

e.g. 2149 => MMCXLIX

Test	Make it red
Code	Make it green
Refactor	Make it right

The Roman Numerals Kata gone wrong demo

Introducing TPP


Robert C Martin - The T
www.youtube.com/watch?v=B93QezwTQpI
YouTube PT
Upload
Sign in
The Transformation Priority Premise
A Workshop
Clean Code
5
NDC 2011 Norwegian Developers Conference
0:10 / 53:43

Robert C Martin - The Transformation Priority Premise

Senad Mehic · 6 videos

Subscribe 8

2,310

21 1

Jim Coplien and Bob Martin Debate TDD
by toalexsmail
7,045
20:59
FEATURED

Clean Code

Test	Make it more specific
Transform	Make it more generic
Refactor	Put on the design hat

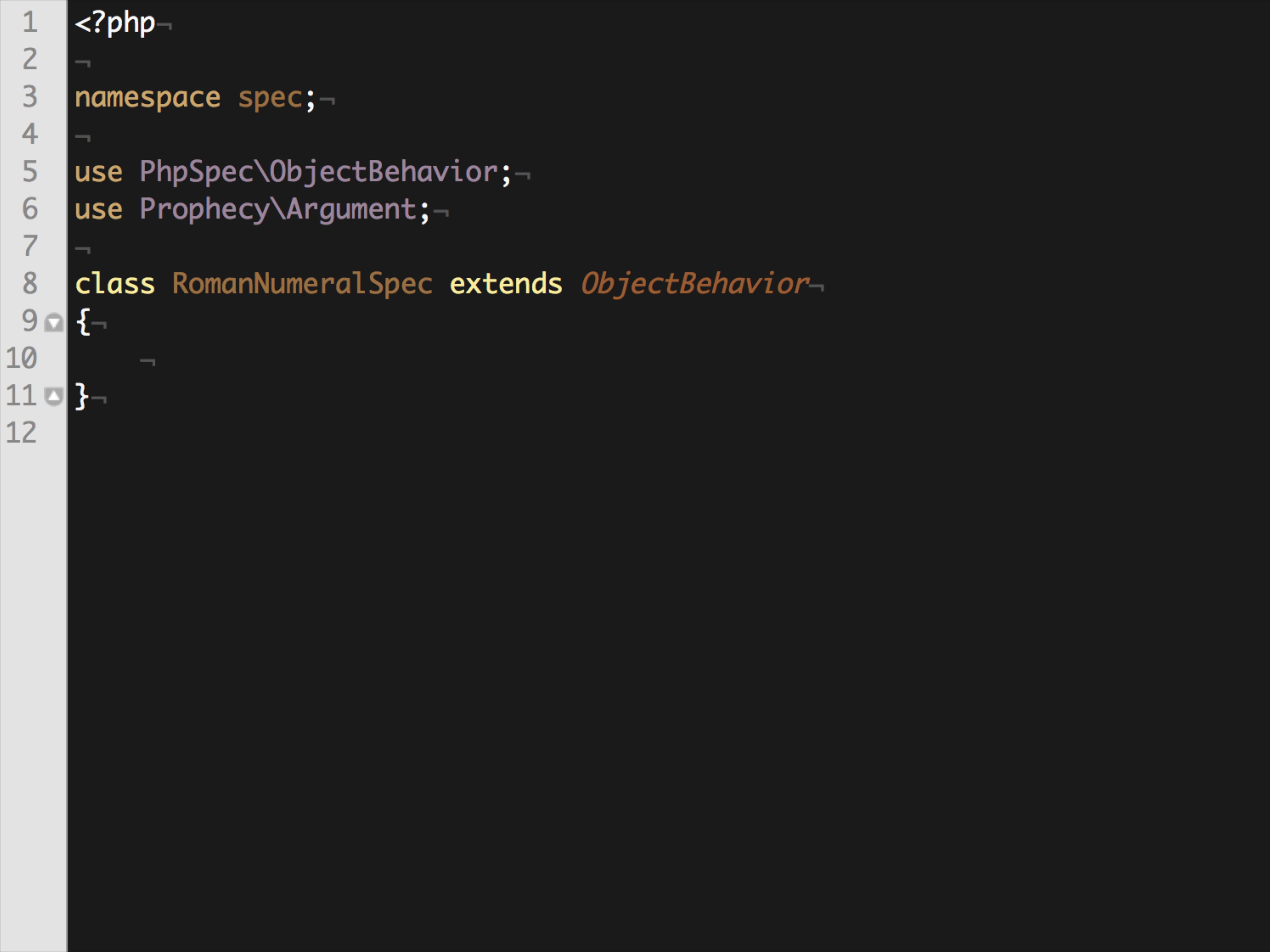
```
null to constant
constant to constant+
constant to scalar
statement to statements
unconditional to if
if to while
```

Transformations (just the most basic ones)

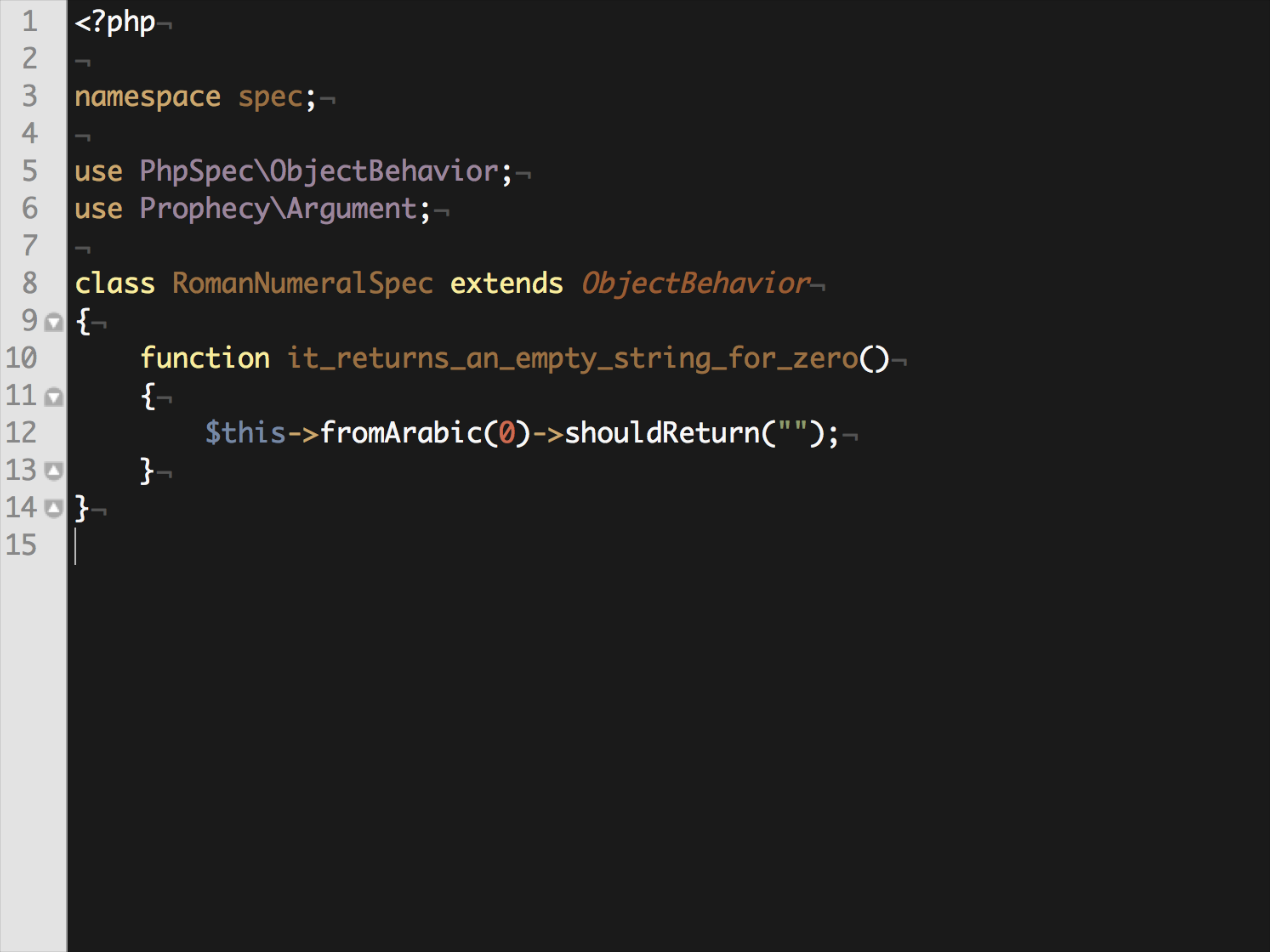
Jim Weirich's Roman Numeral Kata*

* in PHP

http://www.flickr.com/photos/jorendegroof/4471189540/



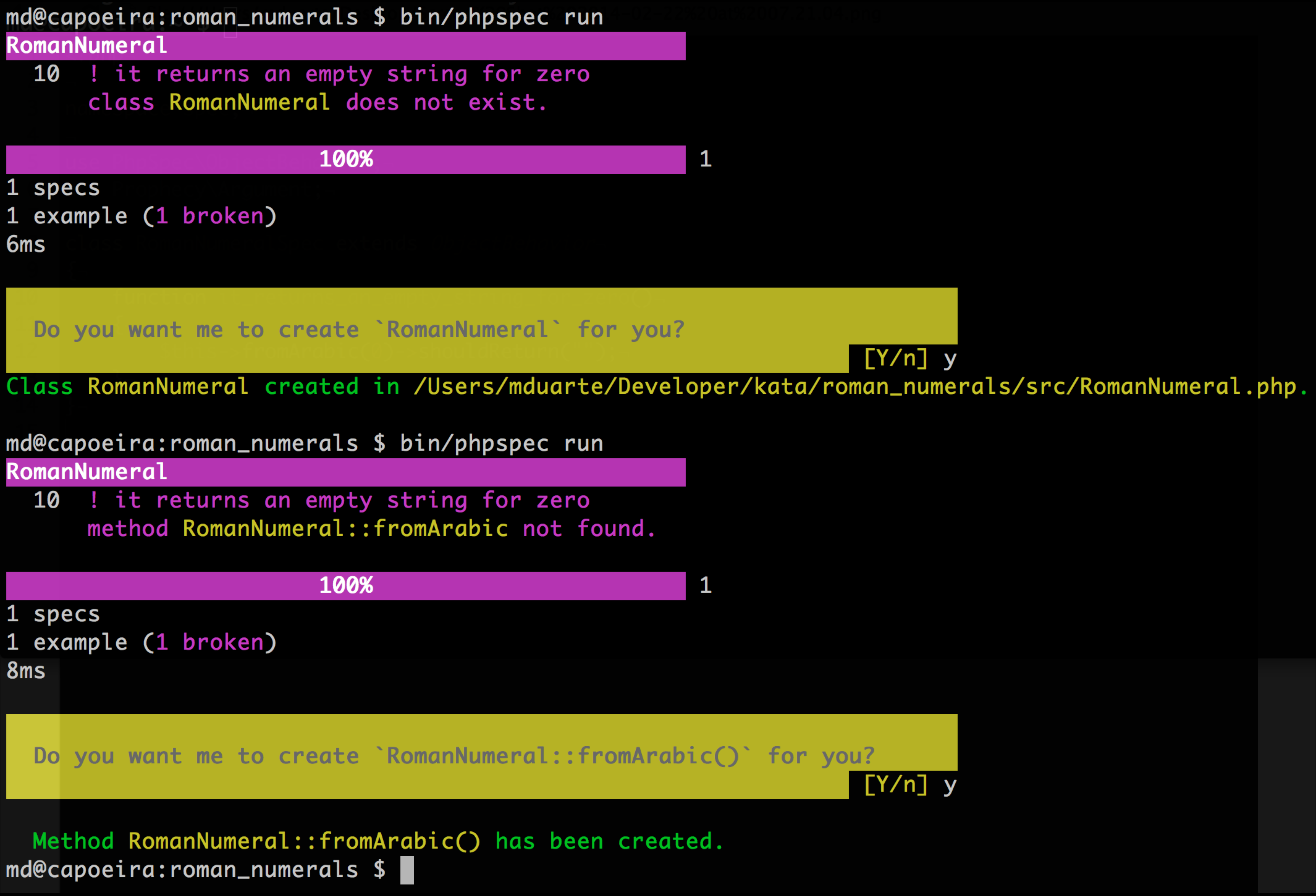
```php
<?php

namespace spec;

use PhpSpec\ObjectBehavior;
use Prophecy\Argument;

class RomanNumeralSpec extends ObjectBehavior
{
    
}
```



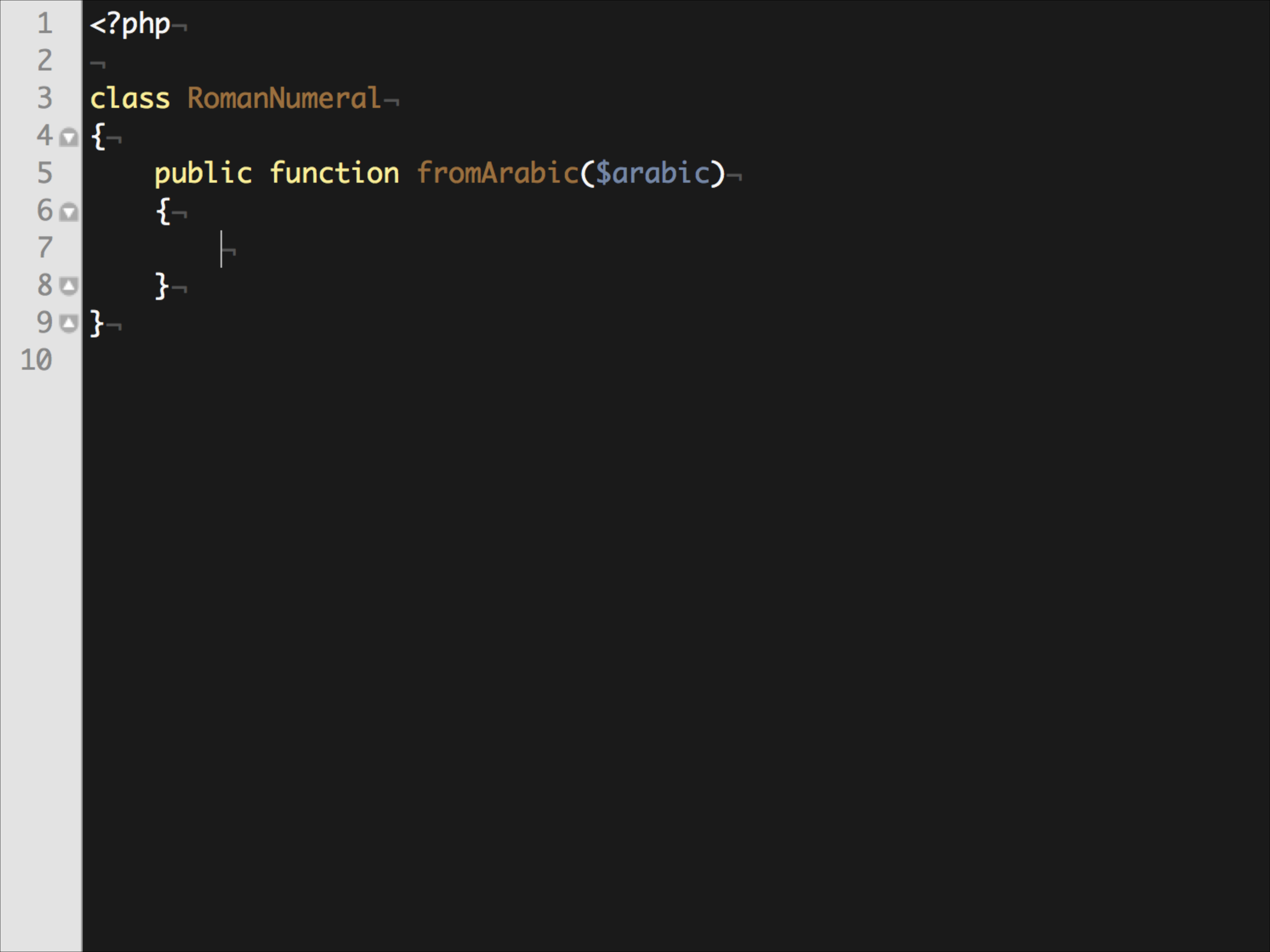
```php
<?php

namespace spec;

use PhpSpec\ObjectBehavior;
use Prophecy\Argument;

class RomanNumeralSpec extends ObjectBehavior
{
    function it_returns_an_empty_string_for_zero()
    {
        $this->fromArabic(0)->shouldReturn("");
    }
}
```

Walking skeleton

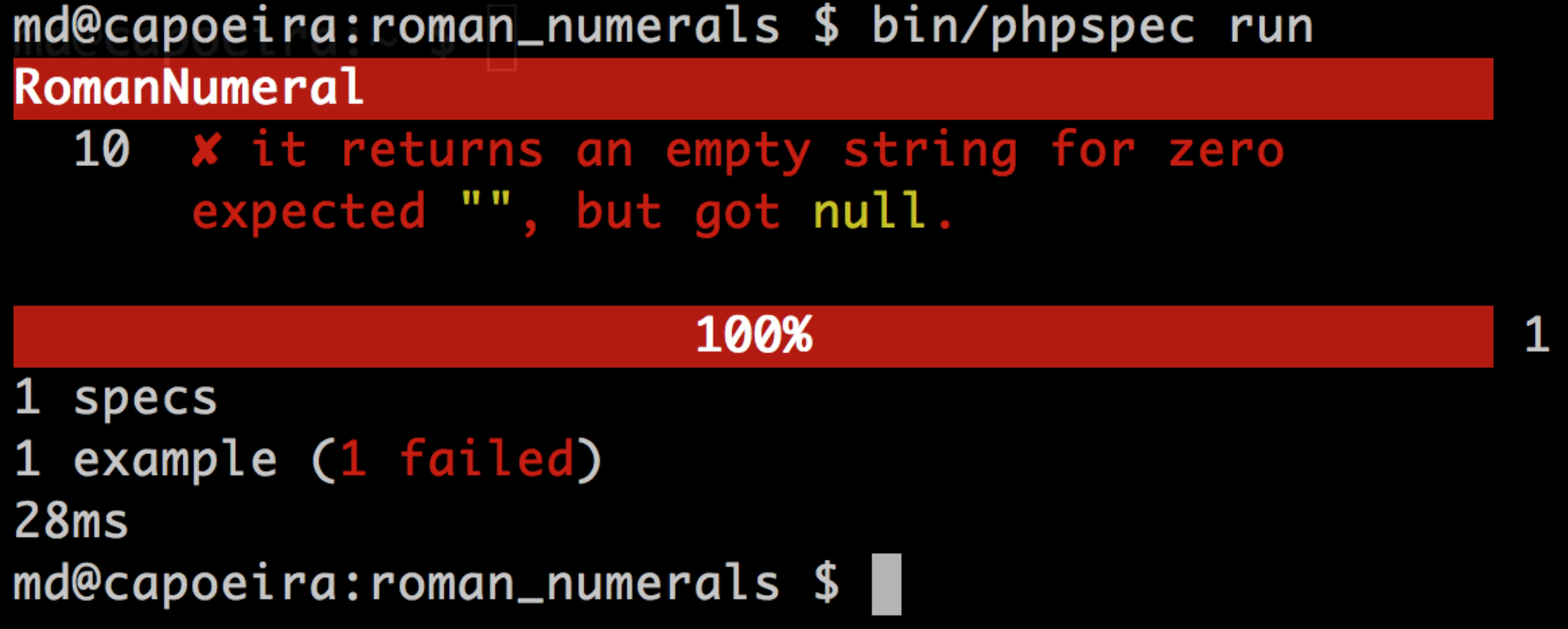
[Cockburn 94]



```
md@capoeira:roman_numerals $ bin/phpspec run
RomanNumeral
  10  ! it returns an empty string for zero
        class RomanNumeral does not exist.

                       100%                        1
1 specs
1 example (1 broken)
6ms


  Do you want me to create `RomanNumeral` for you?
                                                              [Y/n] y
Class RomanNumeral created in /Users/mduarte/Developer/kata/roman_numerals/src/RomanNumeral.php.

md@capoeira:roman_numerals $ bin/phpspec run
RomanNumeral
  10  ! it returns an empty string for zero
        method RomanNumeral::fromArabic not found.

                       100%                        1
1 specs
1 example (1 broken)
8ms


  Do you want me to create `RomanNumeral::fromArabic()` for you?
                                                              [Y/n] y

  Method RomanNumeral::fromArabic() has been created.
md@capoeira:roman_numerals $
```



```php
<?php

class RomanNumeral
{
    public function fromArabic($arabic)
    {
        
    }
}
```

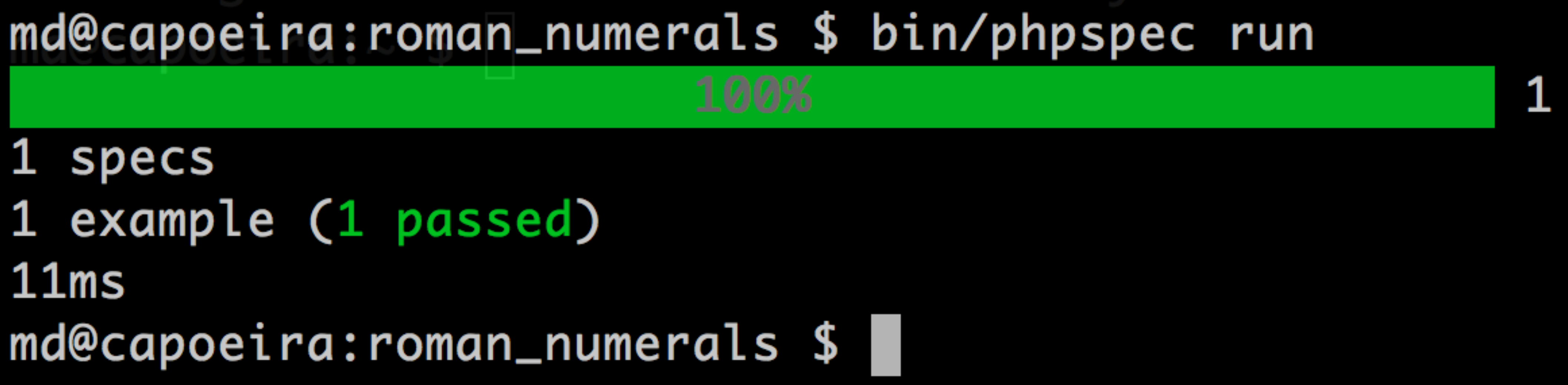


```
md@capoeira:roman_numerals $ bin/phpspec run
RomanNumeral
  10  ✘ it returns an empty string for zero
        expected "", but got null.

                        100%                          1
1 specs
1 example (1 failed)
28ms
md@capoeira:roman_numerals $
```

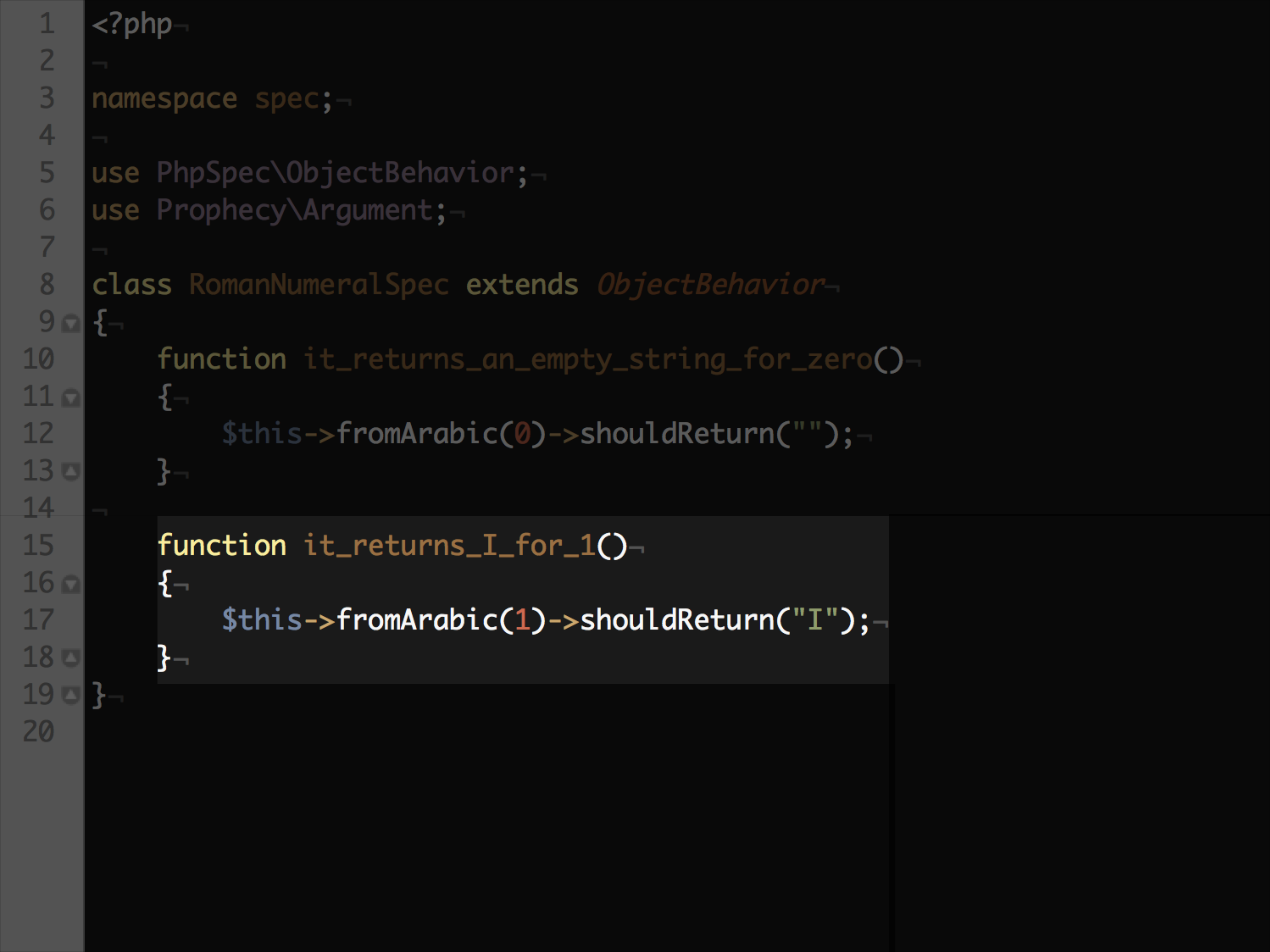
Transformation

`null -> constant`

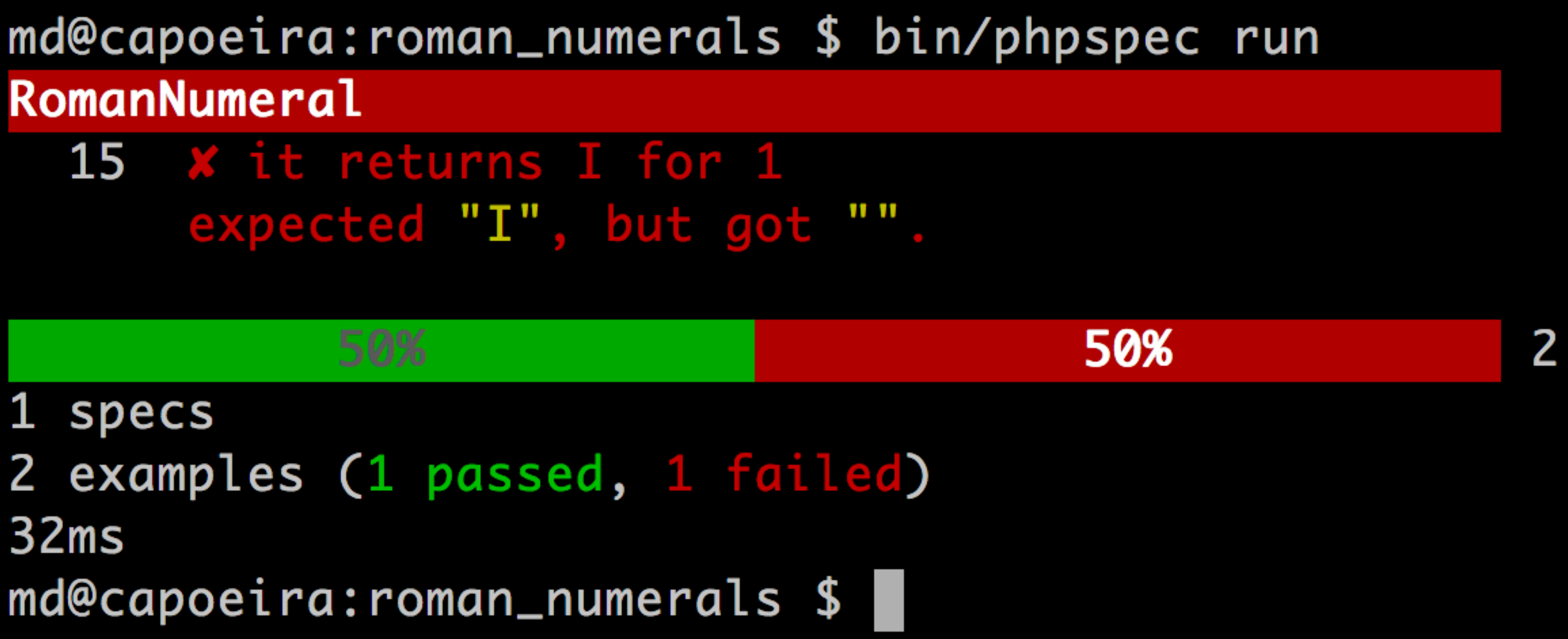
```php
<?php

class RomanNumeral
{
    public function fromArabic($arabic)
    {
        return "";
    }
}
```



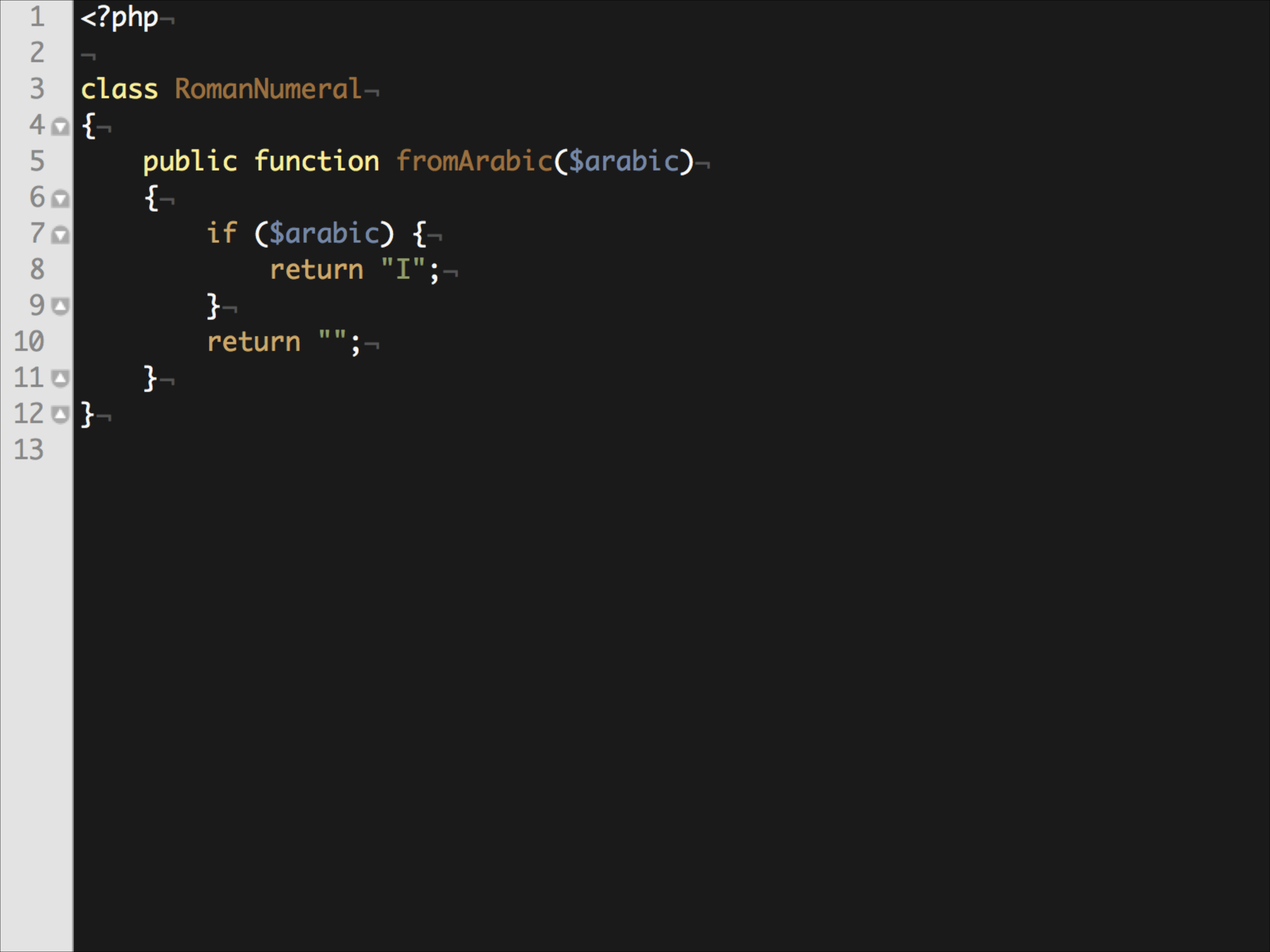
```
md@capoeira:roman_numerals $ bin/phpspec run
                         100%                                  1
1 specs
1 example (1 passed)
11ms
md@capoeira:roman_numerals $ 
```



```php
<?php

namespace spec;

use PhpSpec\ObjectBehavior;
use Prophecy\Argument;

class RomanNumeralSpec extends ObjectBehavior
{
    function it_returns_an_empty_string_for_zero()
    {
        $this->fromArabic(0)->shouldReturn("");
    }

    function it_returns_I_for_1()
    {
        $this->fromArabic(1)->shouldReturn("I");
    }
}
```



```
md@capoeira:roman_numerals $ bin/phpspec run
RomanNumeral
  15  ✘ it returns I for 1
        expected "I", but got "".

                 50%                      50%               2
1 specs
2 examples (1 passed, 1 failed)
32ms
md@capoeira:roman_numerals $ 
```

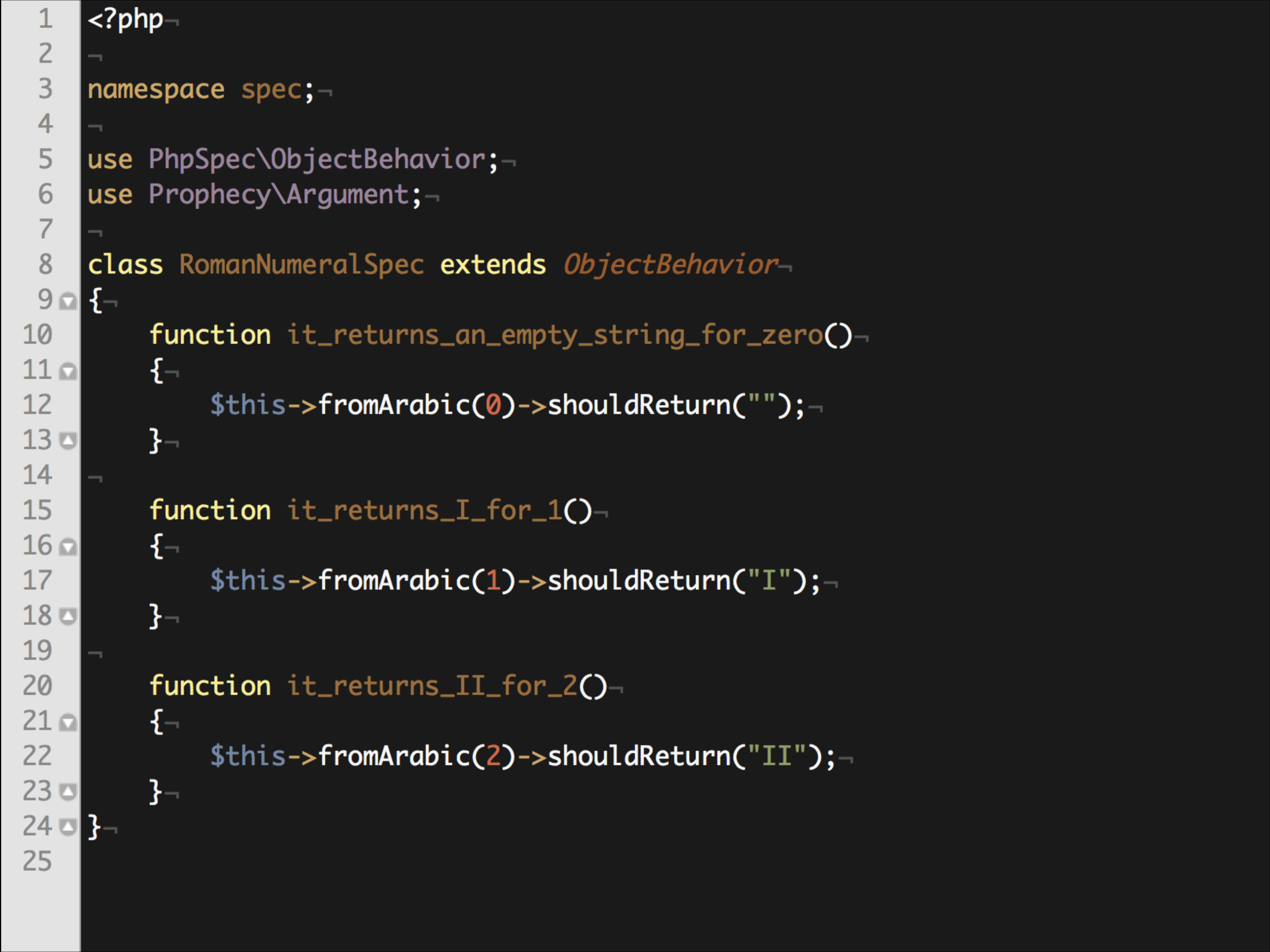


```php
<?php

class RomanNumeral
{
    public function fromArabic($arabic)
    {
        if ($arabic) {
            return "I";
        }
        return "";
    }
}
```

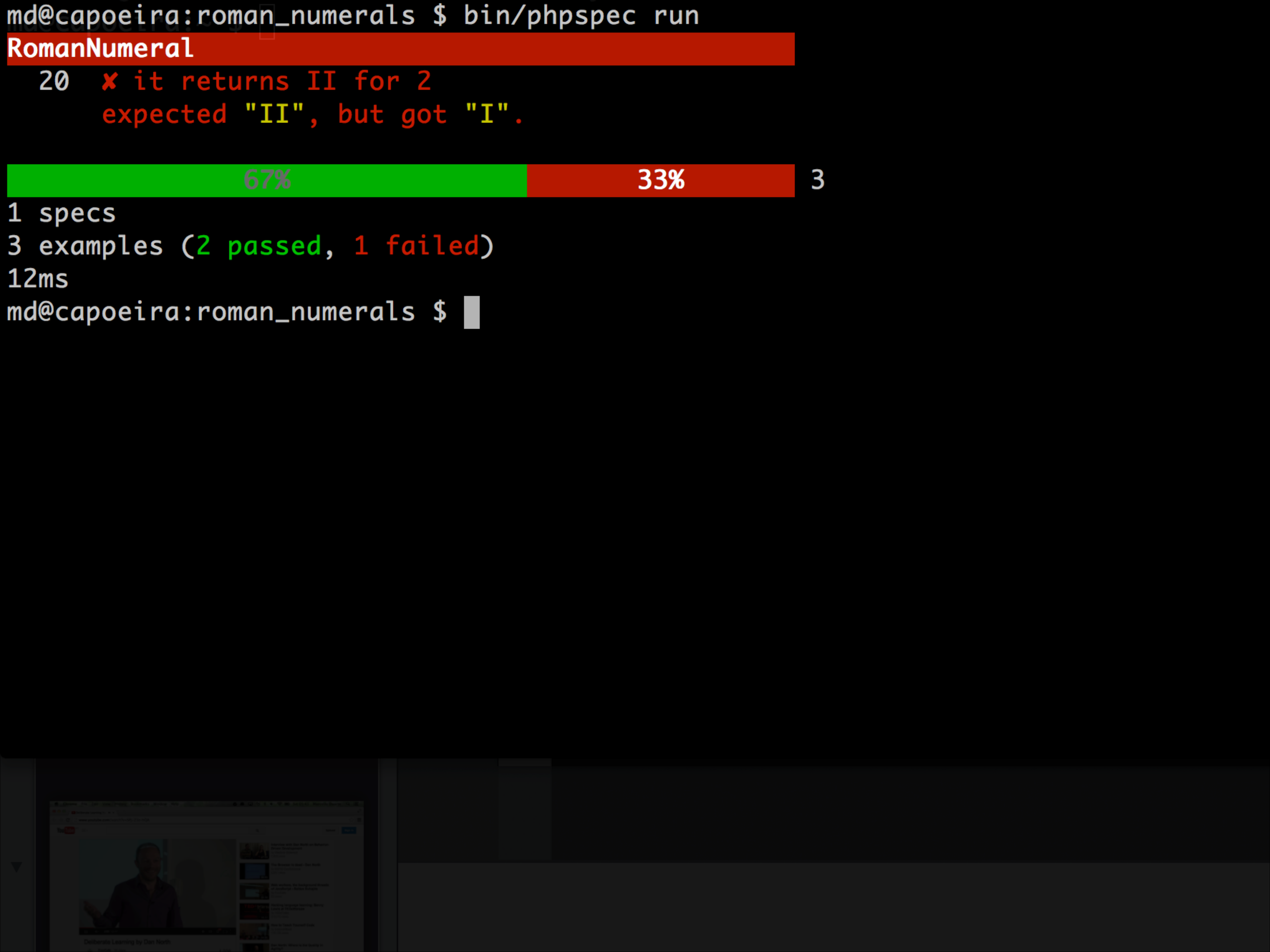
Transformation

`unconditional -> if`

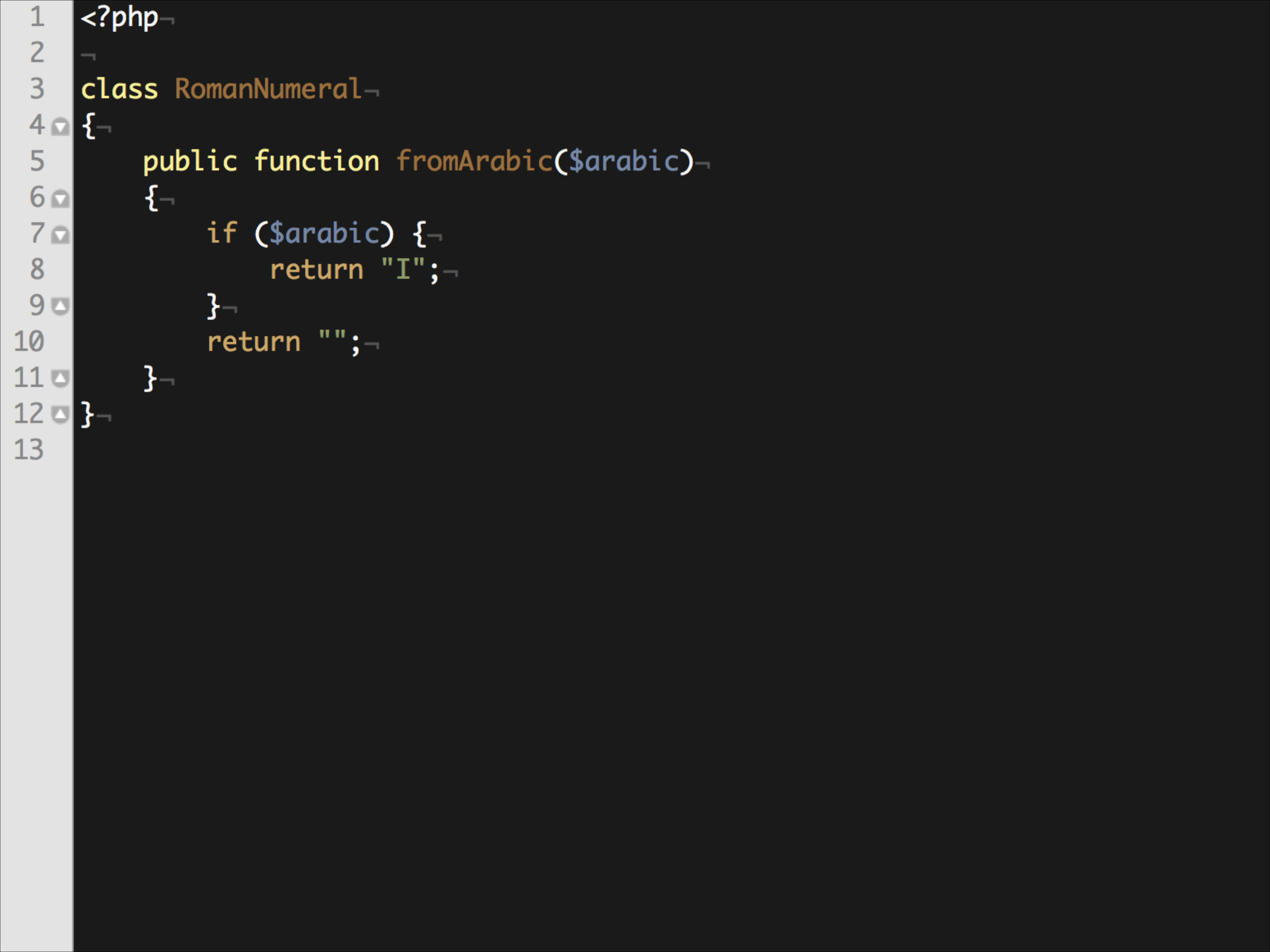
```php
<?php

class RomanNumeral
{
    public function fromArabic($arabic)
    {
        if ($arabic) {
            return "I";
        }
        return "";
    }
}
```



```php
<?php

namespace spec;

use PhpSpec\ObjectBehavior;
use Prophecy\Argument;

class RomanNumeralSpec extends ObjectBehavior
{
    function it_returns_an_empty_string_for_zero()
    {
        $this->fromArabic(0)->shouldReturn("");
    }

    function it_returns_I_for_1()
    {
        $this->fromArabic(1)->shouldReturn("I");
    }

    function it_returns_II_for_2()
    {
        $this->fromArabic(2)->shouldReturn("II");
    }
}
```



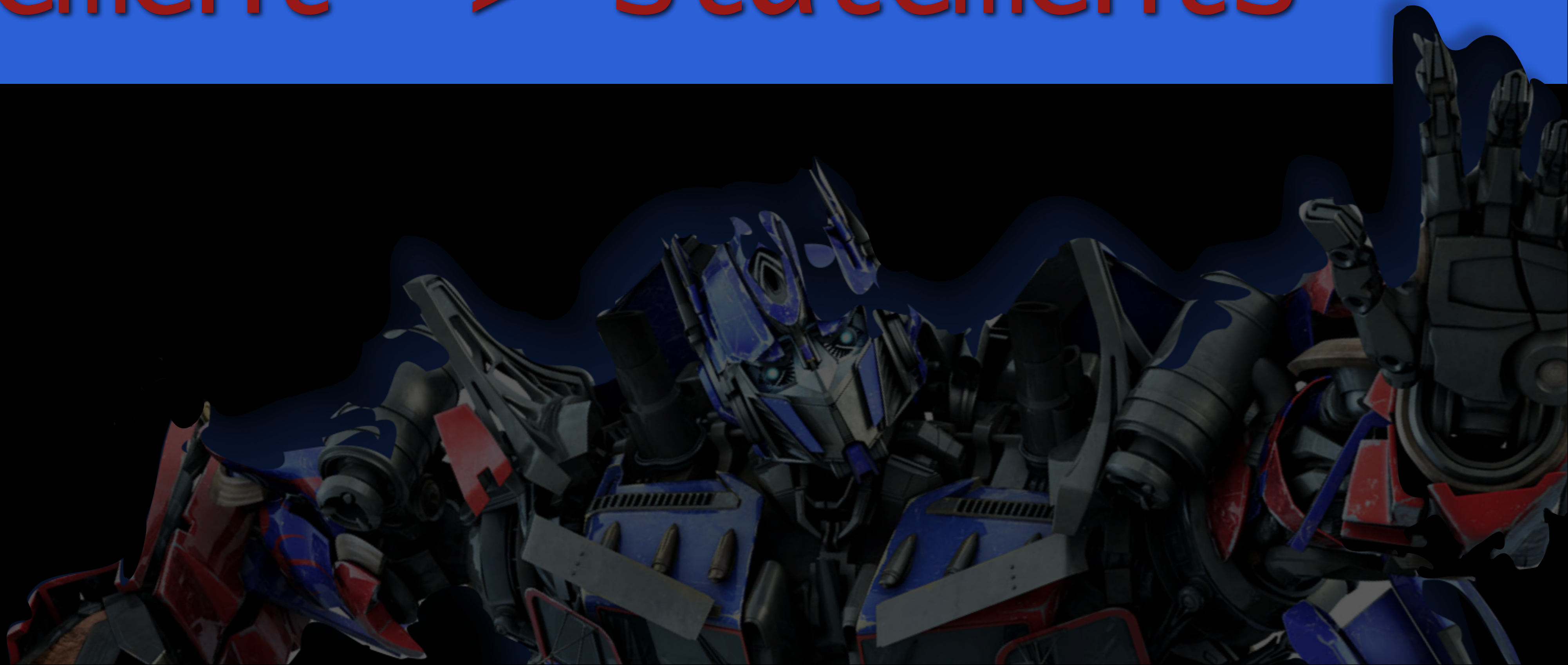
```
md@capoeira:roman_numerals $ bin/phpspec run
RomanNumeral
  20  ✘ it returns II for 2
        expected "II", but got "I".

                 67%                              33%              3
1 specs
3 examples (2 passed, 1 failed)
12ms
md@capoeira:roman_numerals $
```

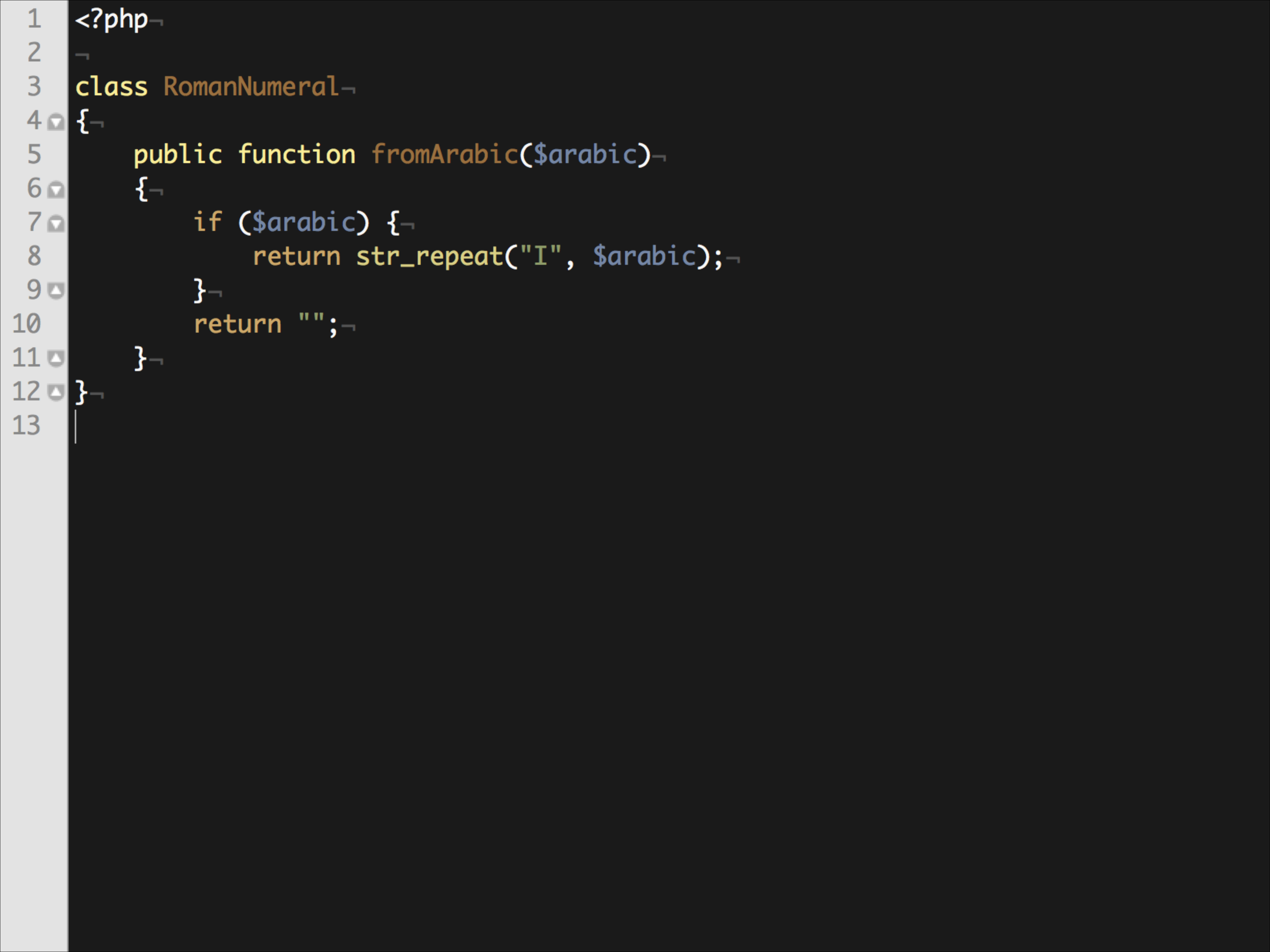


```php
<?php

class RomanNumeral
{
    public function fromArabic($arabic)
    {
        if ($arabic) {
            return "I";
        }
        return "";
    }
}
```

Transformation

statement -> statements





```php
<?php

class RomanNumeral
{
    public function fromArabic($arabic)
    {
        if ($arabic) {
            return str_repeat("I", $arabic);
        }
        return "";
    }
}

```

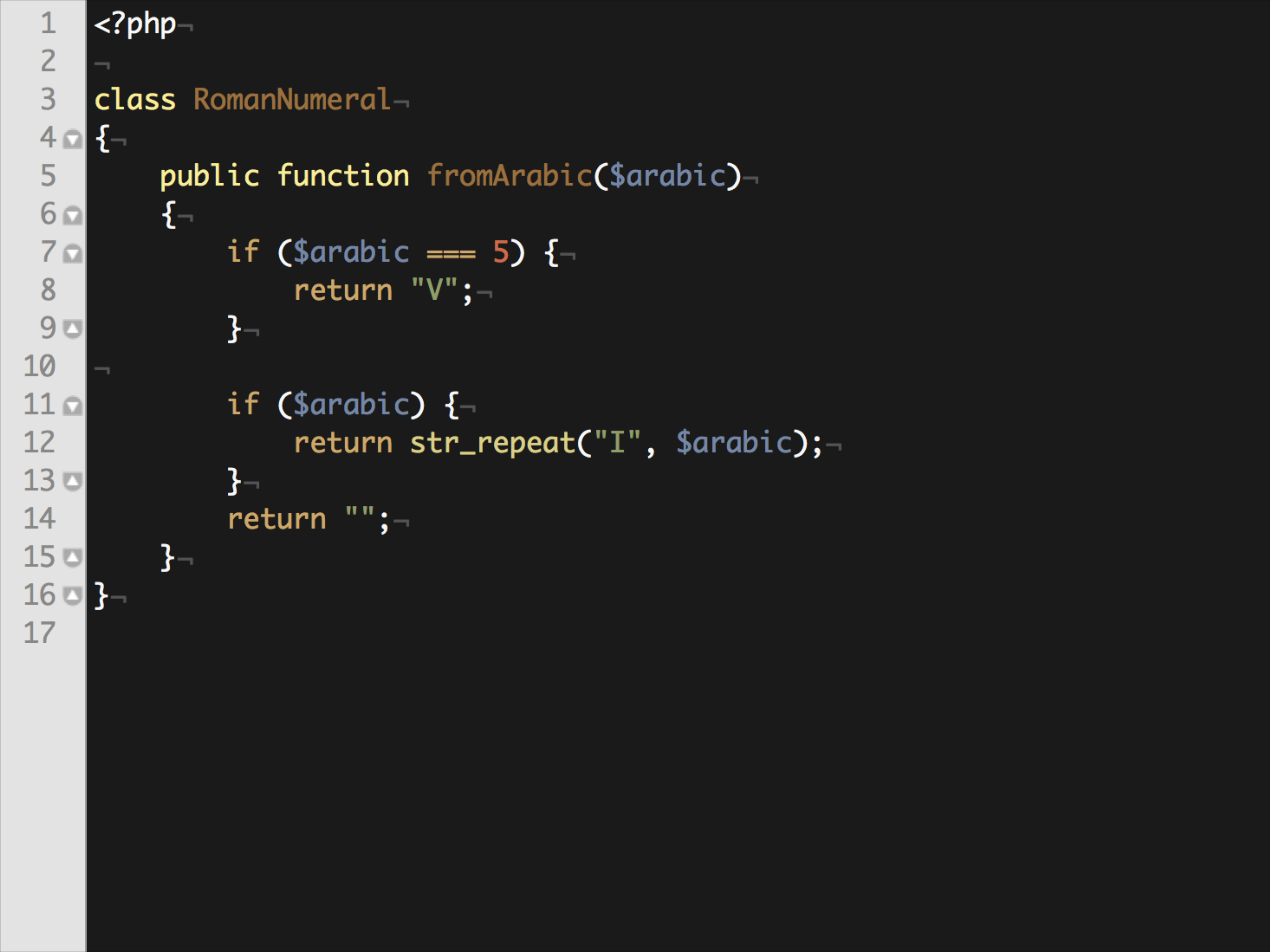
```php

class RomanNumeralSpec extends ObjectBehavior
{
    function it_returns_an_empty_string_for_zero()
    {
        $this->fromArabic(0)->shouldReturn("");
    }

    function it_returns_I_for_1()
    {
        $this->fromArabic(1)->shouldReturn("I");
    }

    function it_returns_II_for_2()
    {
        $this->fromArabic(2)->shouldReturn("II");
    }

    function it_returns_V_for_5()
    {
        $this->fromArabic(5)->shouldReturn("V");
    }
}

```

```
md@capoeira:roman_numerals $ bin/phpspec run
RomanNumeral
  25  ✘ it returns V for 5
        expected "V", but got "IIIII".

                   75%                      25%       4
1 specs
4 examples (3 passed, 1 failed)
11ms
md@capoeira:roman_numerals $
```

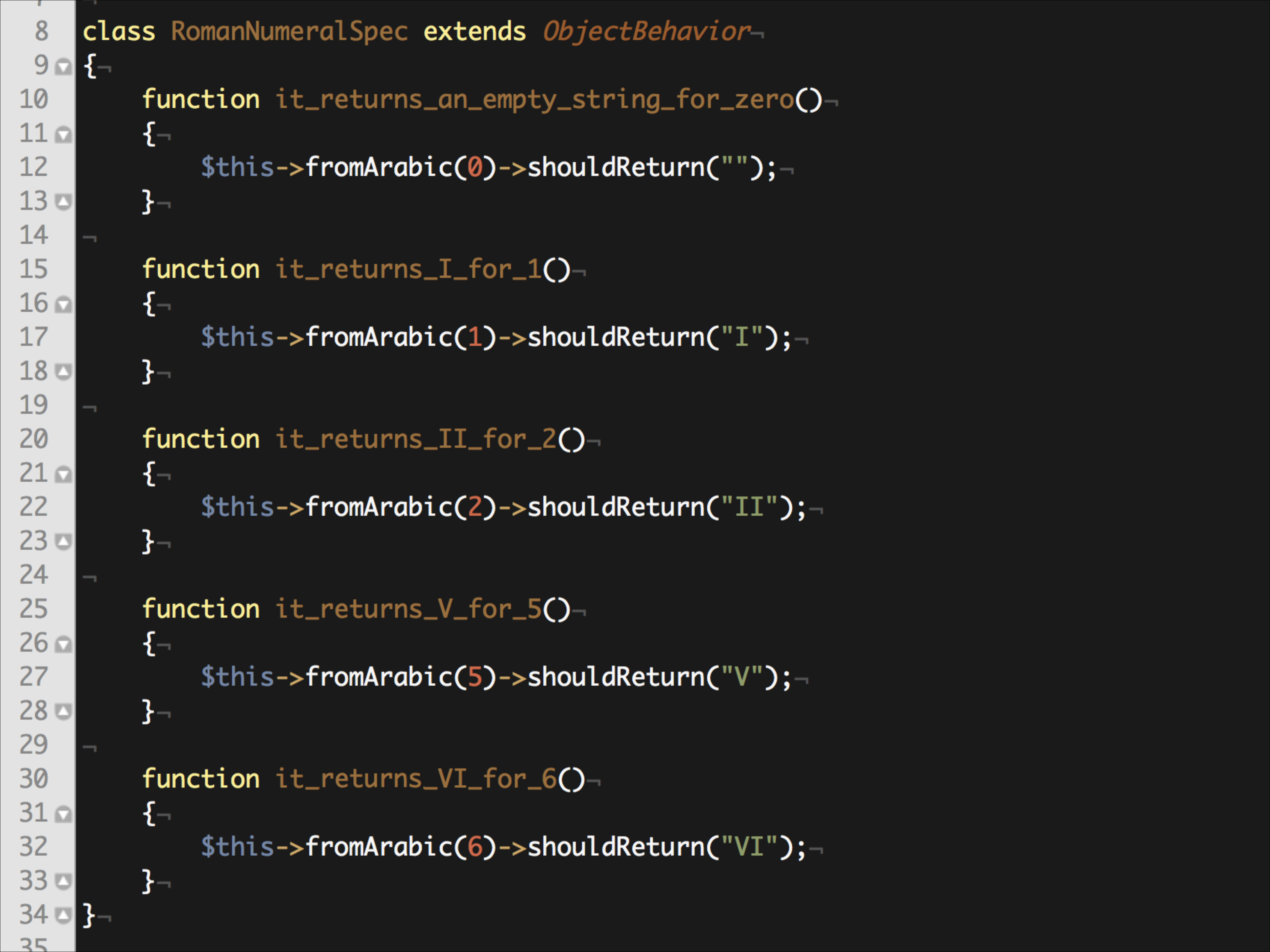
```php
<?php

class RomanNumeral
{
    public function fromArabic($arabic)
    {
        if ($arabic) {
            return str_repeat("I", $arabic);
        }
        return "";
    }
}

```

Transformation

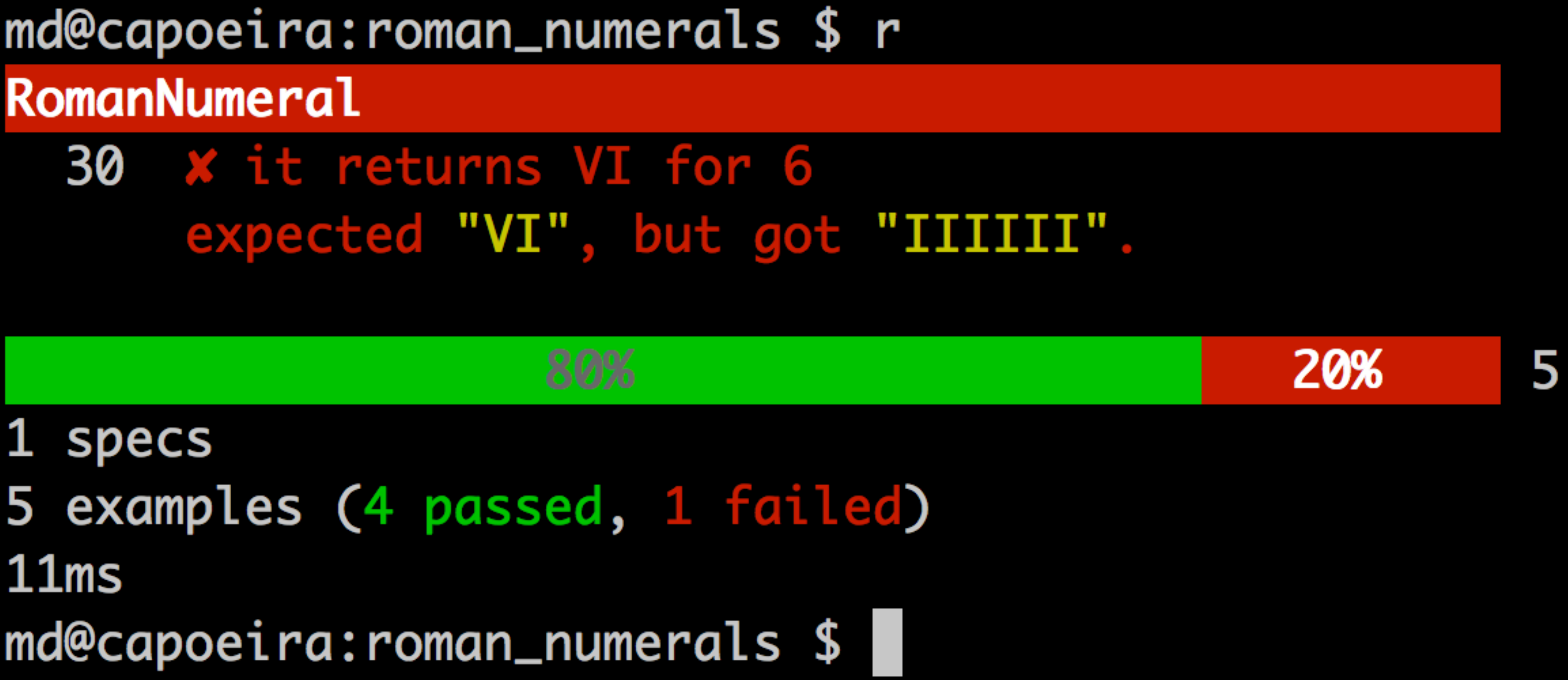
`unconditional -> if`



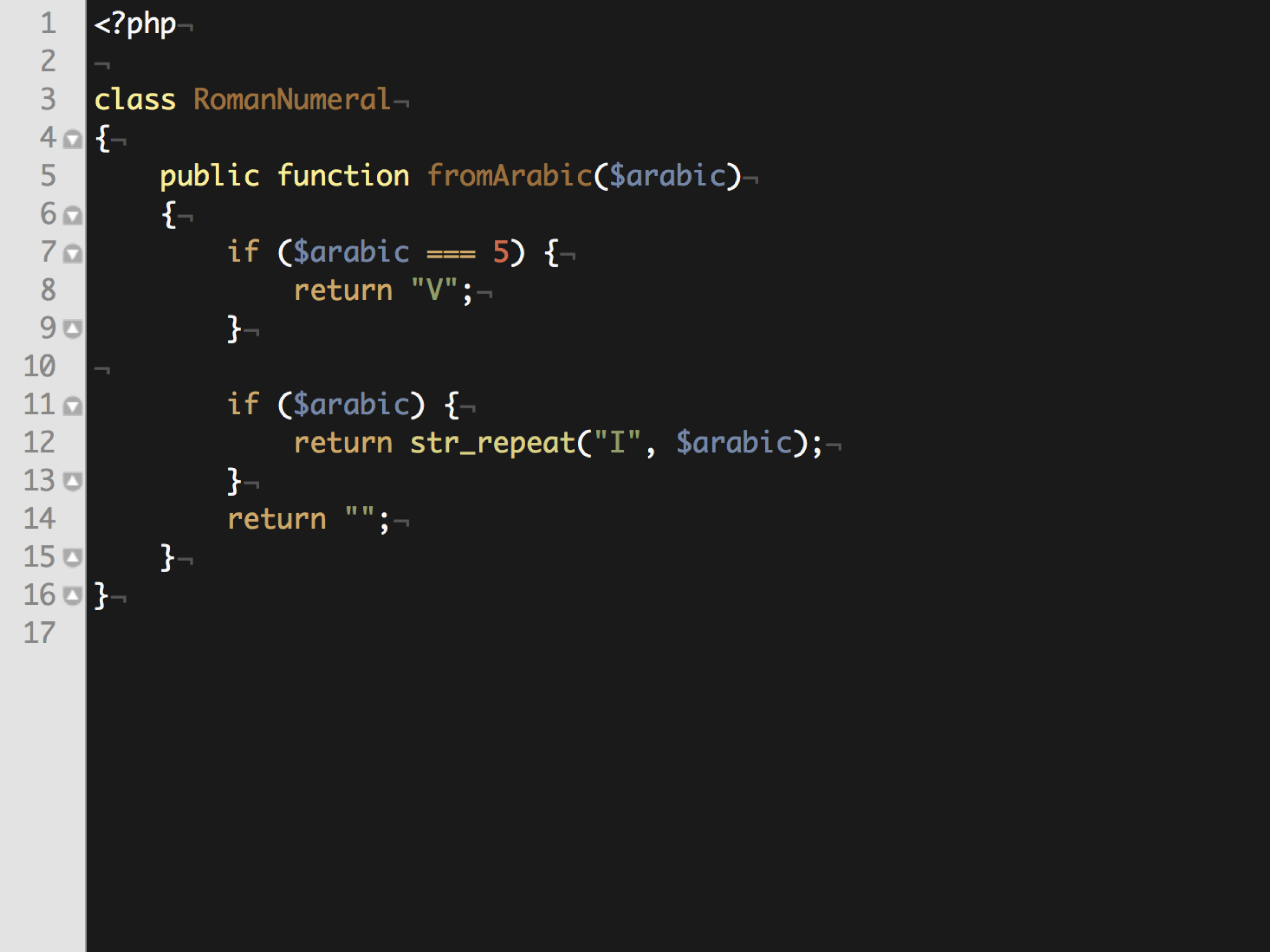
```php
<?php

class RomanNumeral
{
    public function fromArabic($arabic)
    {
        if ($arabic === 5) {
            return "V";
        }

        if ($arabic) {
            return str_repeat("I", $arabic);
        }
        return "";
    }
}
```



```php
class RomanNumeralSpec extends ObjectBehavior
{
    function it_returns_an_empty_string_for_zero()
    {
        $this->fromArabic(0)->shouldReturn("");
    }

    function it_returns_I_for_1()
    {
        $this->fromArabic(1)->shouldReturn("I");
    }

    function it_returns_II_for_2()
    {
        $this->fromArabic(2)->shouldReturn("II");
    }

    function it_returns_V_for_5()
    {
        $this->fromArabic(5)->shouldReturn("V");
    }

    function it_returns_VI_for_6()
    {
        $this->fromArabic(6)->shouldReturn("VI");
    }
}
```

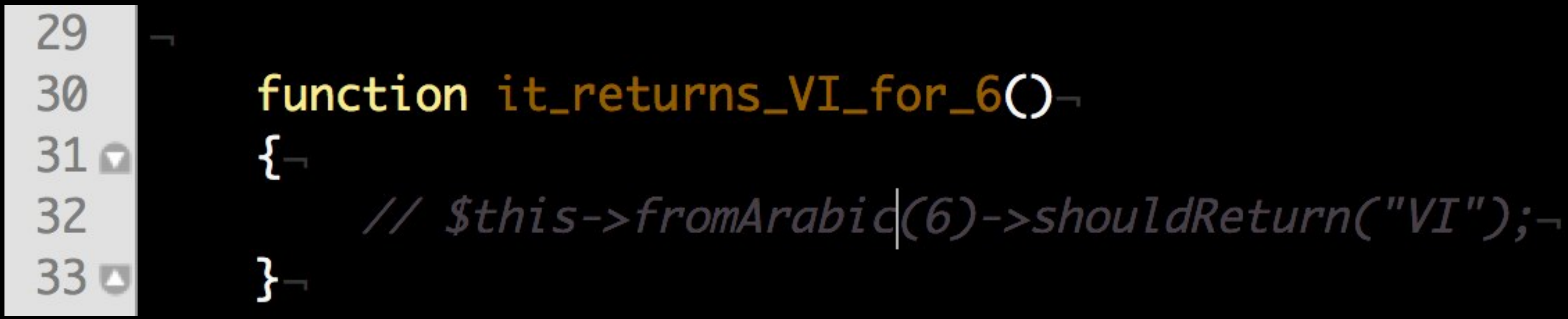


```
md@capoeira:roman_numerals $ r
RomanNumeral
  30  ✘ it returns VI for 6
        expected "VI", but got "IIIIII".

                    80%                         20%     5
1 specs
5 examples (4 passed, 1 failed)
11ms
md@capoeira:roman_numerals $
```

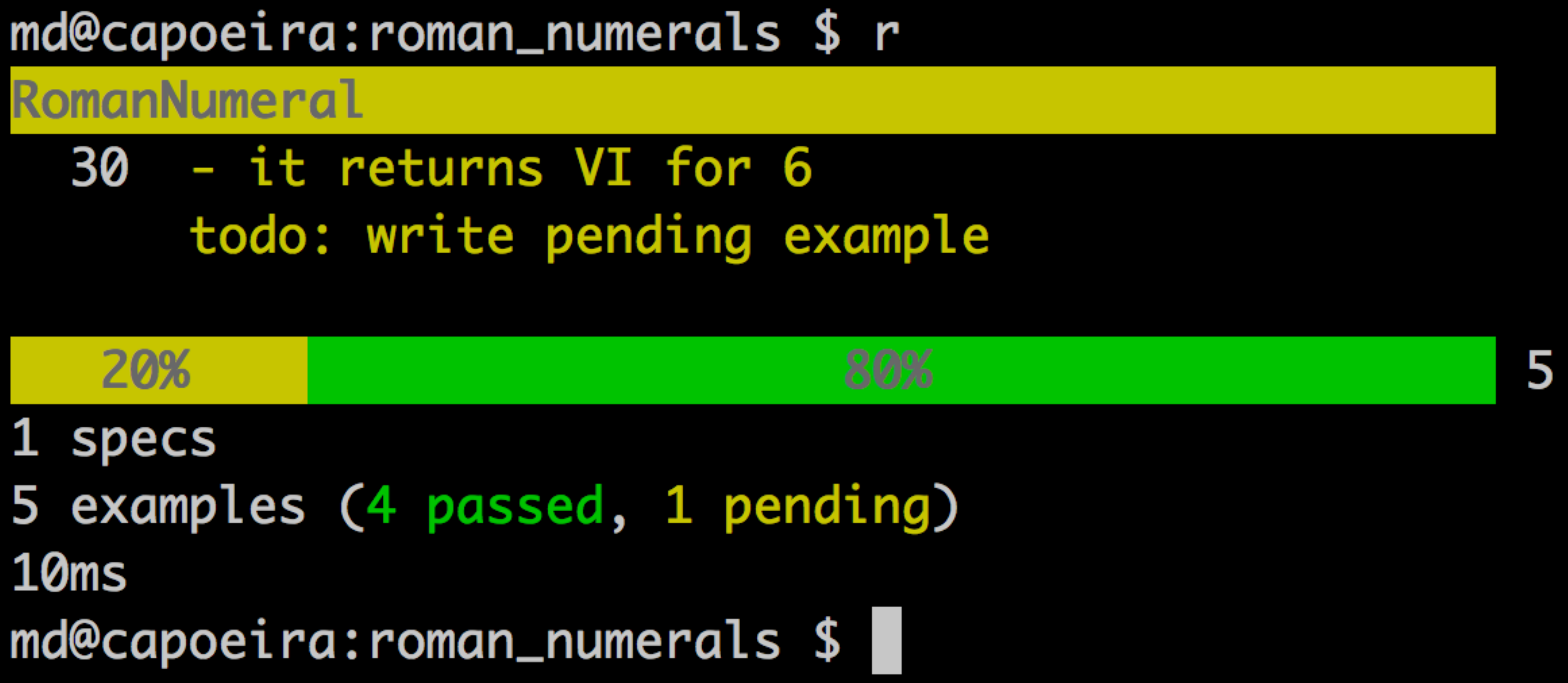


```php
<?php

class RomanNumeral
{
    public function fromArabic($arabic)
    {
        if ($arabic === 5) {
            return "V";
        }

        if ($arabic) {
            return str_repeat("I", $arabic);
        }
        return "";
    }
}
```

Refactor time?



```php

    function it_returns_VI_for_6()
    {
        // $this->fromArabic(6)->shouldReturn("VI");
    }
```

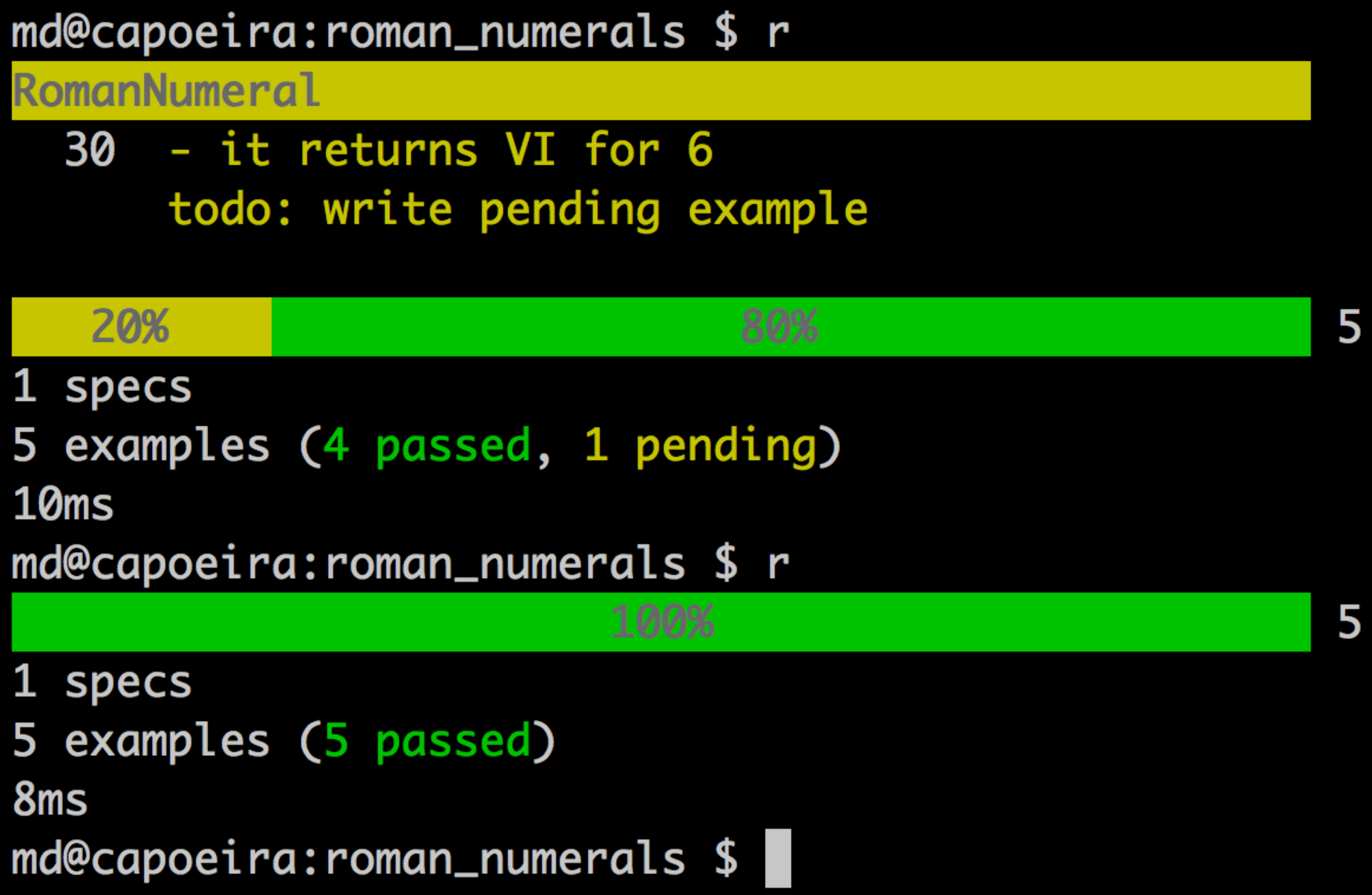


```
md@capoeira:roman_numerals $ r
RomanNumeral
  30  - it returns VI for 6
      todo: write pending example

   20%               80%                       5
1 specs
5 examples (4 passed, 1 pending)
10ms
md@capoeira:roman_numerals $
```

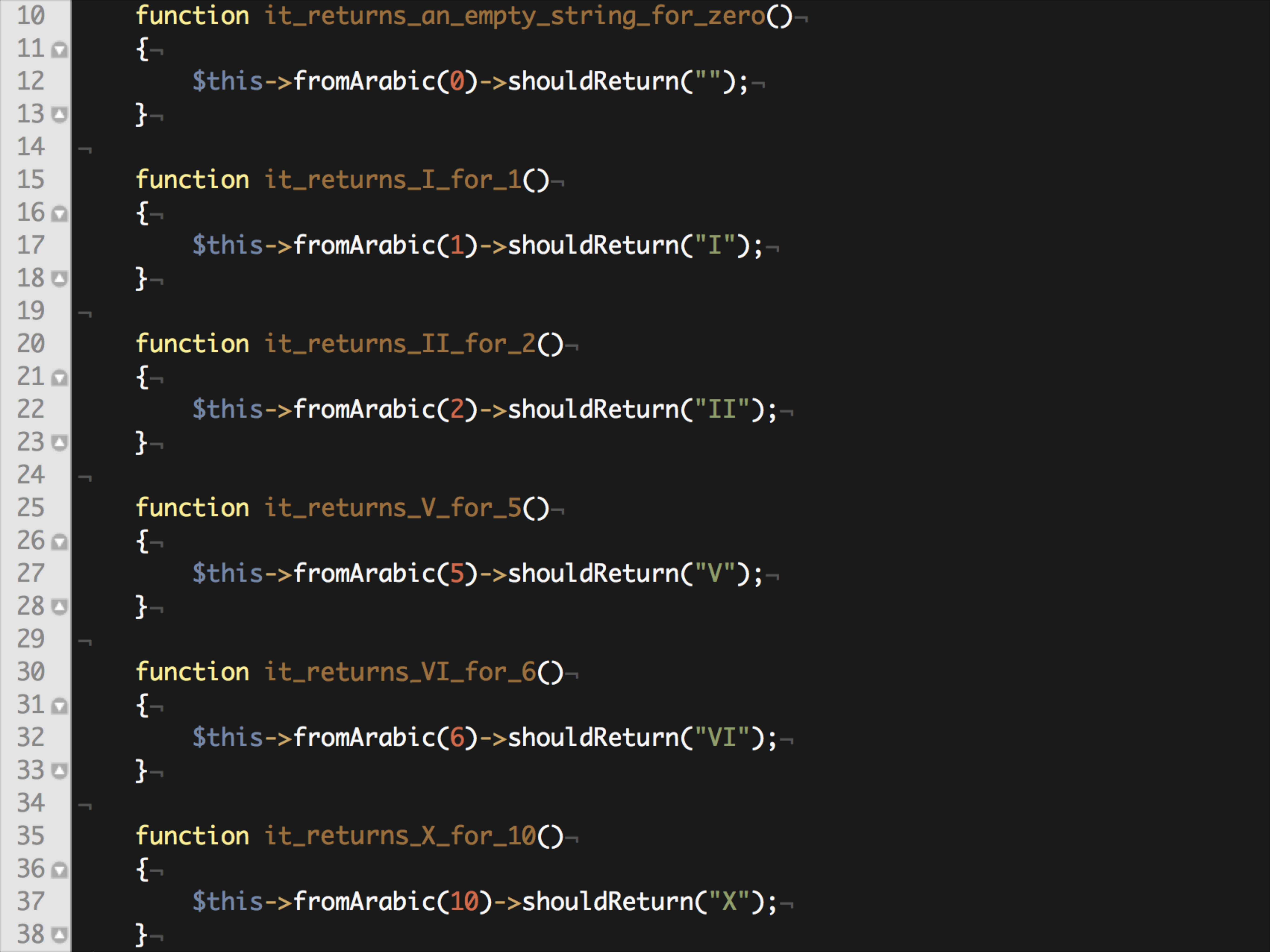
```php
<?php

class RomanNumeral
{
    public function fromArabic($arabic)
    {
        if ($arabic === 5) {
            return "V";
        }

        if ($arabic) {
            return str_repeat("I", $arabic);
        }
        return "";
    }
}
```

```php
<?php

class RomanNumeral
{
    public function fromArabic($arabic)
    {
        $roman = "";

        if ($arabic === 5) {
            $roman = "V";
            $arabic -= 5;
        }

        if ($arabic) {
            $roman .=  str_repeat("I", $arabic);
        }
        return $roman;
    }
}
```

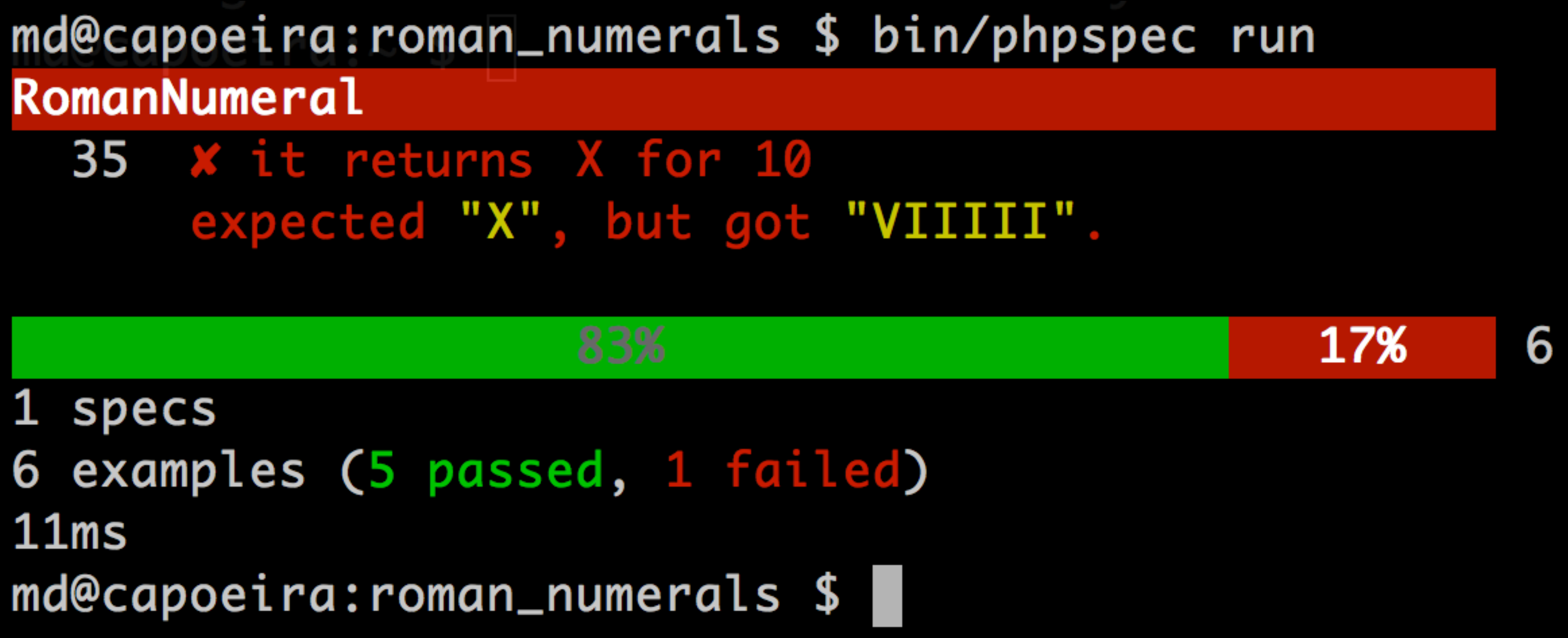
```php
class RomanNumeralSpec extends ObjectBehavior
{
    function it_returns_an_empty_string_for_zero()
    {
        $this->fromArabic(0)->shouldReturn("");
    }

    function it_returns_I_for_1()
    {
        $this->fromArabic(1)->shouldReturn("I");
    }

    function it_returns_II_for_2()
    {
        $this->fromArabic(2)->shouldReturn("II");
    }

    function it_returns_V_for_5()
    {
        $this->fromArabic(5)->shouldReturn("V");
    }

    function it_returns_VI_for_6()
    {
        $this->fromArabic(6)->shouldReturn("VI");
    }
}
```



```
md@capoeira:roman_numerals $ r
RomanNumeral
  30  - it returns VI for 6
      todo: write pending example

    20%                    80%                   5
1 specs
5 examples (4 passed, 1 pending)
10ms
md@capoeira:roman_numerals $ r
                       100%                      5
1 specs
5 examples (5 passed)
8ms
md@capoeira:roman_numerals $
```

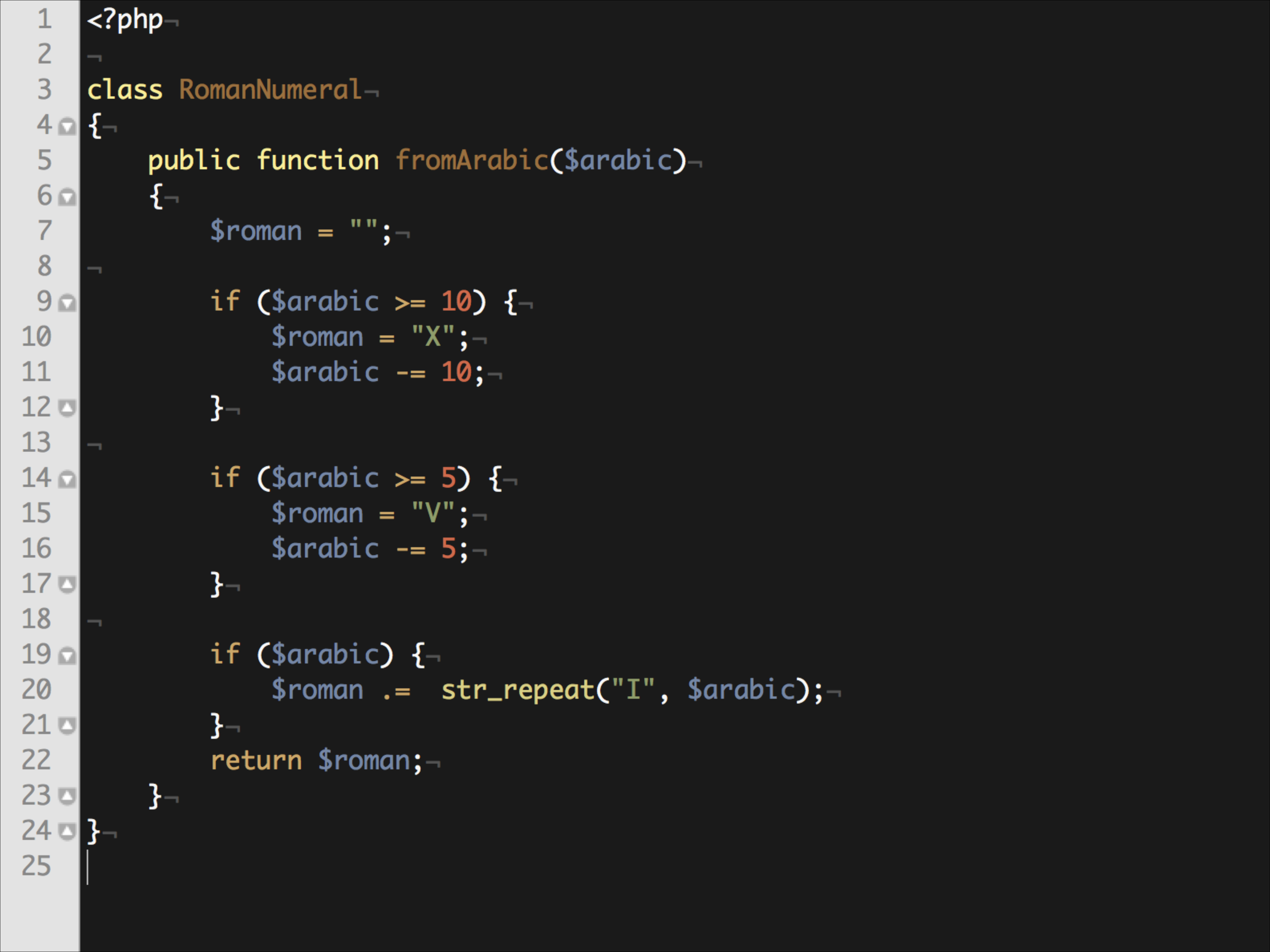


```php
    function it_returns_an_empty_string_for_zero()
    {
        $this->fromArabic(0)->shouldReturn("");
    }

    function it_returns_I_for_1()
    {
        $this->fromArabic(1)->shouldReturn("I");
    }

    function it_returns_II_for_2()
    {
        $this->fromArabic(2)->shouldReturn("II");
    }

    function it_returns_V_for_5()
    {
        $this->fromArabic(5)->shouldReturn("V");
    }

    function it_returns_VI_for_6()
    {
        $this->fromArabic(6)->shouldReturn("VI");
    }

    function it_returns_X_for_10()
    {
        $this->fromArabic(10)->shouldReturn("X");
    }
```

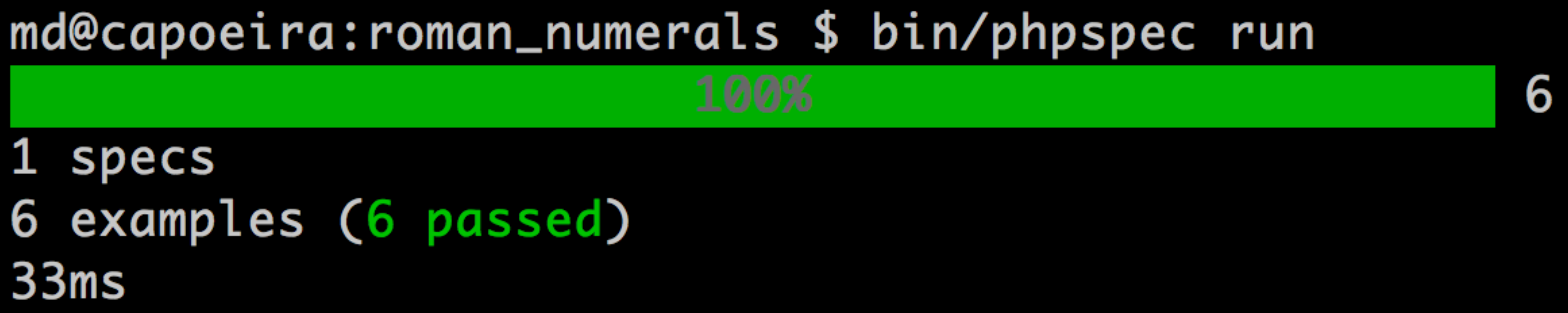


```
md@capoeira:roman_numerals $ bin/phpspec run
RomanNumeral
  35  ✘ it returns X for 10
        expected "X", but got "VIIIII".

                    83%                      17%      6
1 specs
6 examples (5 passed, 1 failed)
11ms
md@capoeira:roman_numerals $ 
```

```php
<?php

class RomanNumeral
{
    public function fromArabic($arabic)
    {
        $roman = "";

        if ($arabic === 5) {
            $roman = "V";
            $arabic -= 5;
        }

        if ($arabic) {
            $roman .=  str_repeat("I", $arabic);
        }
        return $roman;
    }
}

```

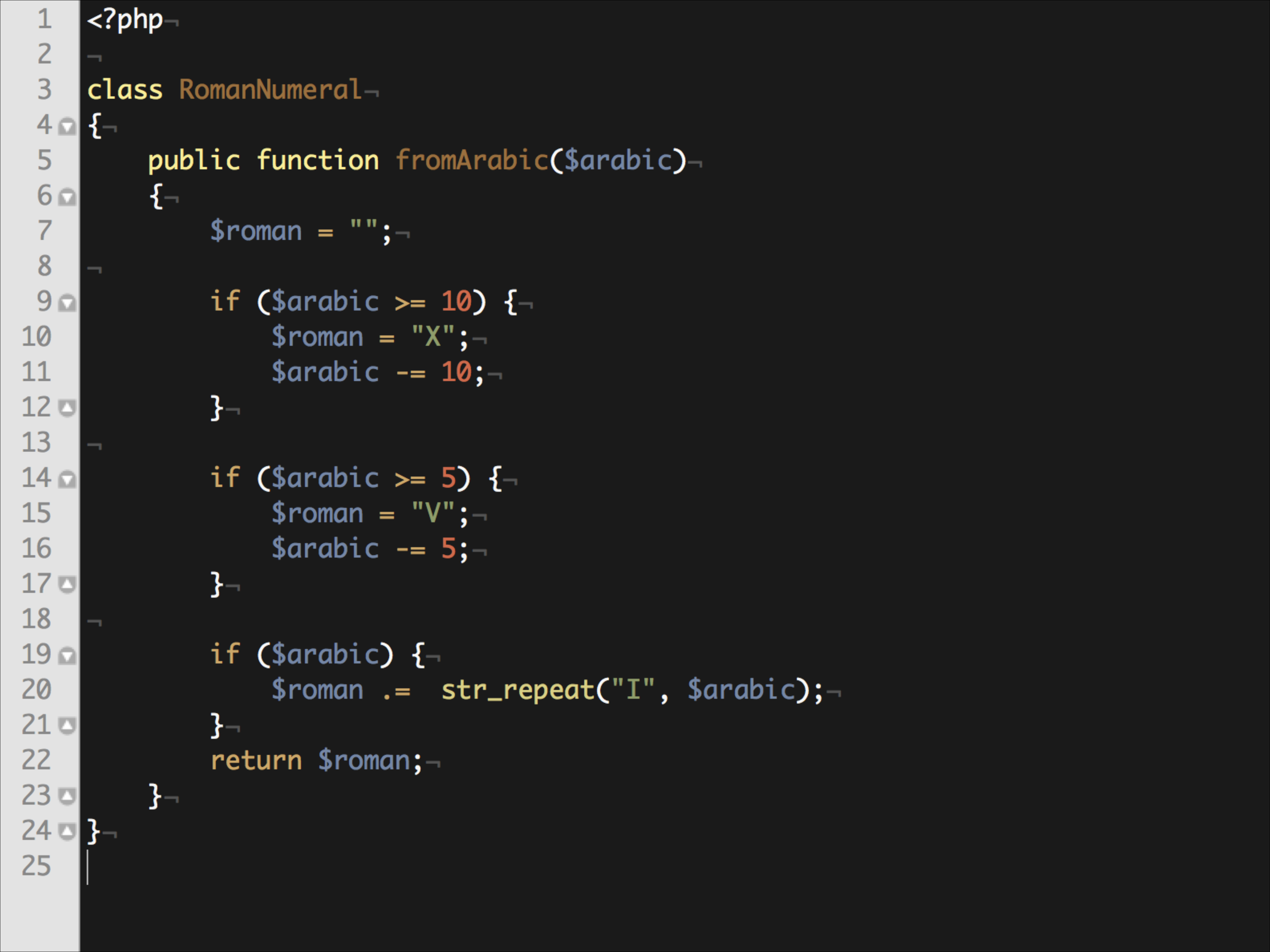


```php
<?php

class RomanNumeral
{
    public function fromArabic($arabic)
    {
        $roman = "";

        if ($arabic >= 10) {
            $roman = "X";
            $arabic -= 10;
        }

        if ($arabic >= 5) {
            $roman = "V";
            $arabic -= 5;
        }

        if ($arabic) {
            $roman .=  str_repeat("I", $arabic);
        }
        return $roman;
    }
}
```


```
md@capoeira:roman_numerals $ bin/phpspec run
                        100%                         6
1 specs
6 examples (6 passed)
33ms
```

```php
    }

    function it_returns_II_for_2()
    {
        $this->fromArabic(2)->shouldReturn("II");
    }

    function it_returns_V_for_5()
    {
        $this->fromArabic(5)->shouldReturn("V");
    }

    function it_returns_VI_for_6()
    {
        $this->fromArabic(6)->shouldReturn("VI");
    }

    function it_returns_X_for_10()
    {
        $this->fromArabic(10)->shouldReturn("X");
    }

    function it_returns_XX_for_20()
    {
        $this->fromArabic(20)->shouldReturn("XX");
    }
}
```

```
md@capoeira:roman_numerals $ bin/phpspec run
RomanNumeral
  40  ✘ it returns XX for 20
        expected "XX", but got "VIIIII".

                    86%                         14%   7
1 specs
7 examples (6 passed, 1 failed)
12ms
md@capoeira:roman_numerals $ 
```



```php
<?php

class RomanNumeral
{
    public function fromArabic($arabic)
    {
        $roman = "";

        if ($arabic >= 10) {
            $roman = "X";
            $arabic -= 10;
        }

        if ($arabic >= 5) {
            $roman = "V";
            $arabic -= 5;
        }

        if ($arabic) {
            $roman .=  str_repeat("I", $arabic);
        }
        return $roman;
    }
}
```

Transformation

`if -> while`

Condition to loop transformation

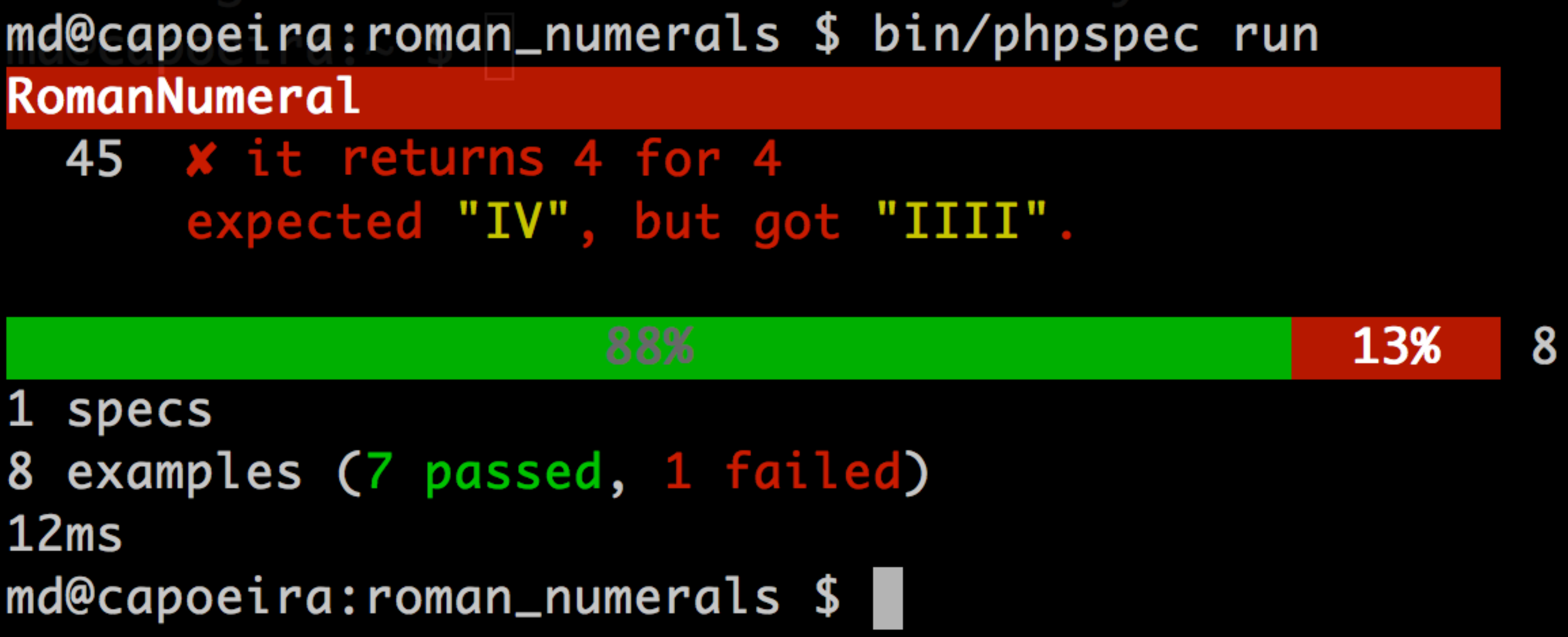
is a great indicator you are in the right track of solving the algorithm

[Martin 13]

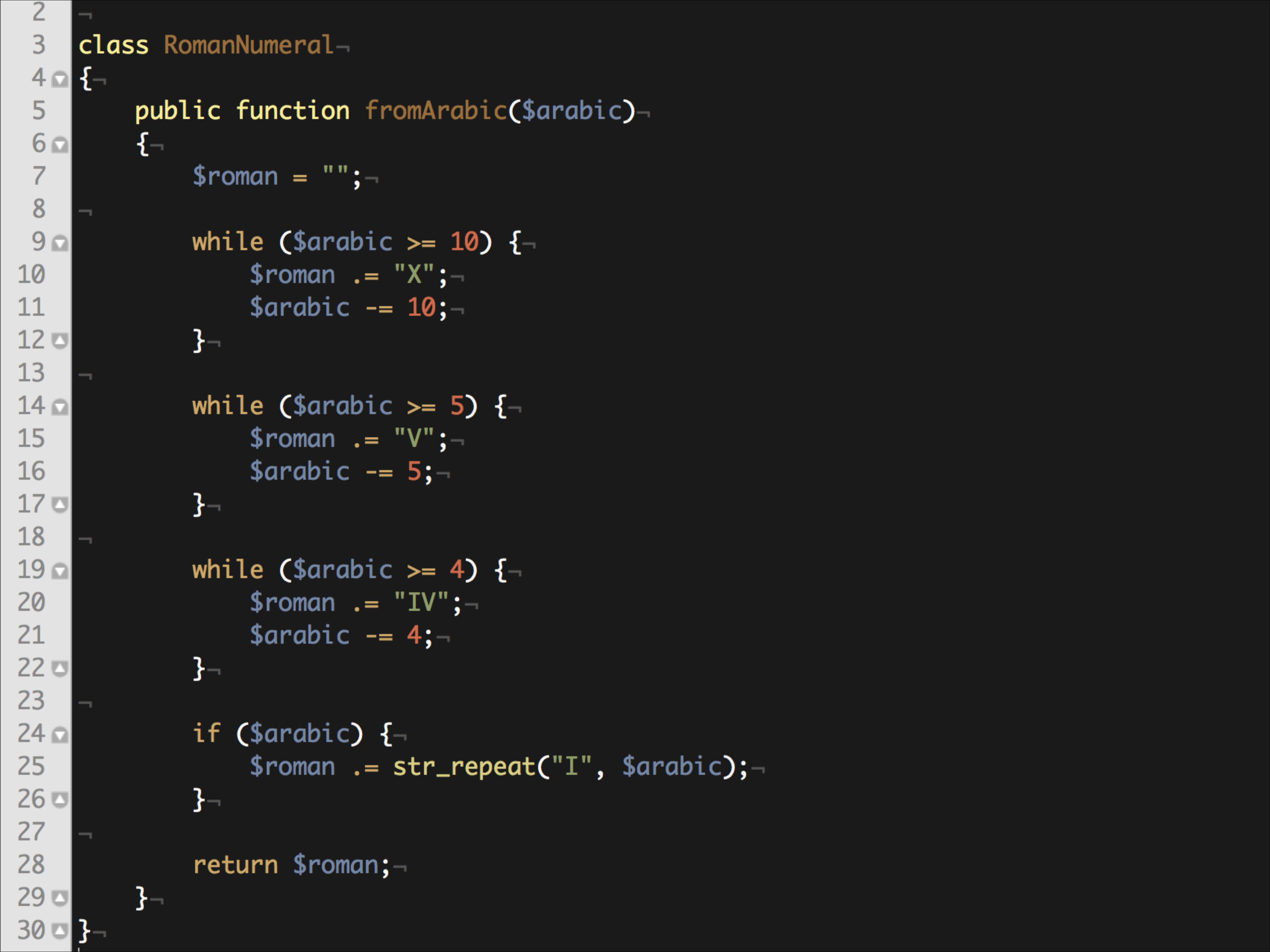
```php
<?php

class RomanNumeral
{
    public function fromArabic($arabic)
    {
        $roman = "";

        while ($arabic >= 10) {
            $roman .= "X";
            $arabic -= 10;
        }

        if ($arabic >= 5) {
            $roman = "V";
            $arabic -= 5;
        }

        if ($arabic) {
            $roman .=  str_repeat("I", $arabic);
        }
        return $roman;
    }
}
```

```
md@capoeira:roman_numerals $ bin/phpspec run
                          100%                          7
1 specs
7 examples (7 passed)
9ms
md@capoeira:roman_numerals $ 
```

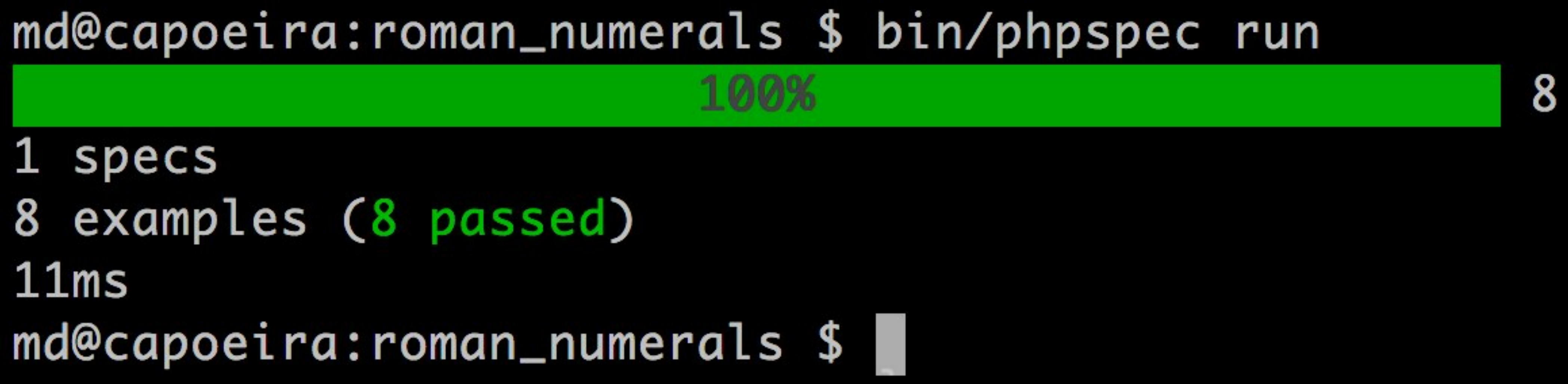
```php
        $this->fromArabic(2)->shouldReturn("II");
    }

    function it_returns_V_for_5()
    {
        $this->fromArabic(5)->shouldReturn("V");
    }

    function it_retunrs_VI_for_6()
    {
        $this->fromArabic(6)->shouldReturn("VI");
    }

    function it_retunrs_X_for_10()
    {
        $this->fromArabic(10)->shouldReturn("X");
    }

    function it_retunrs_XX_for_20()
    {
        $this->fromArabic(20)->shouldReturn("XX");
    }

    function it_retunrs_4_for_4()
    {
        $this->fromArabic(4)->shouldReturn("IV");
    }
}
```



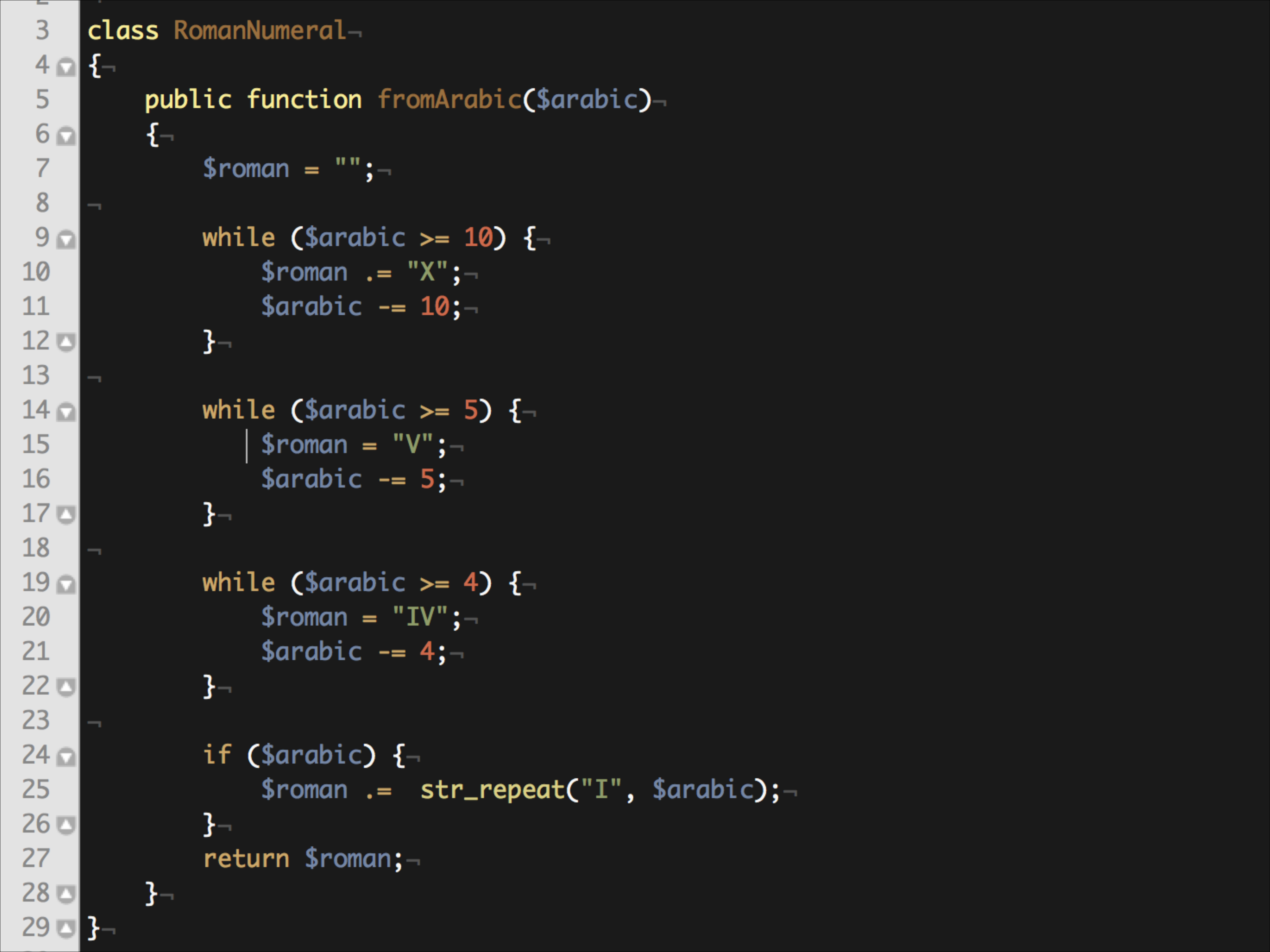
```
md@capoeira:roman_numerals $ bin/phpspec run
RomanNumeral
  45  ✘ it returns 4 for 4
        expected "IV", but got "IIII".

                        88%                          13%   8
1 specs
8 examples (7 passed, 1 failed)
12ms
md@capoeira:roman_numerals $
```



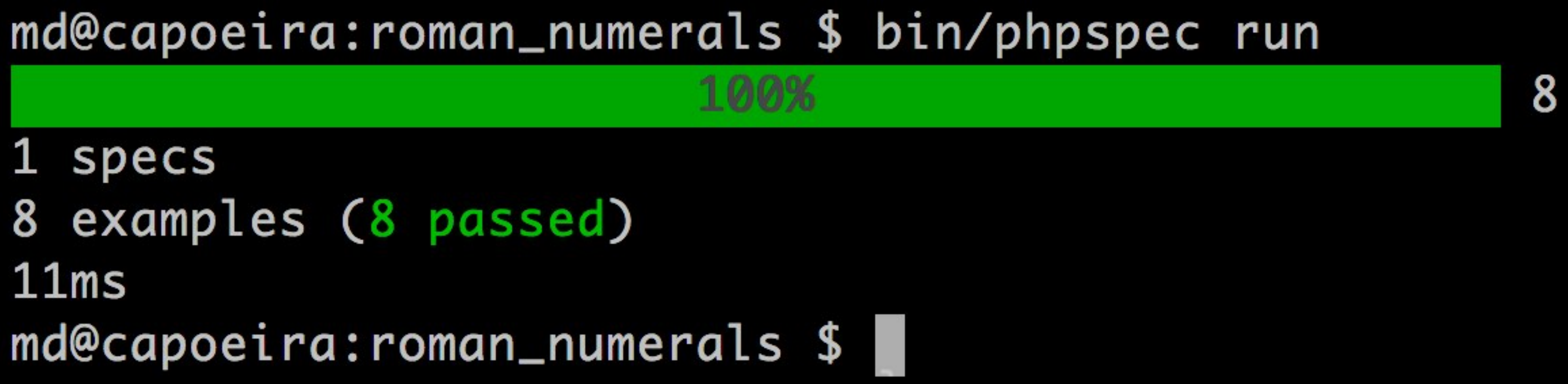
```php

class RomanNumeral
{
    public function fromArabic($arabic)
    {
        $roman = "";

        while ($arabic >= 10) {
            $roman .= "X";
            $arabic -= 10;
        }

        while ($arabic >= 5) {
            $roman .= "V";
            $arabic -= 5;
        }

        while ($arabic >= 4) {
            $roman .= "IV";
            $arabic -= 4;
        }

        if ($arabic) {
            $roman .= str_repeat("I", $arabic);
        }

        return $roman;
    }
}
```



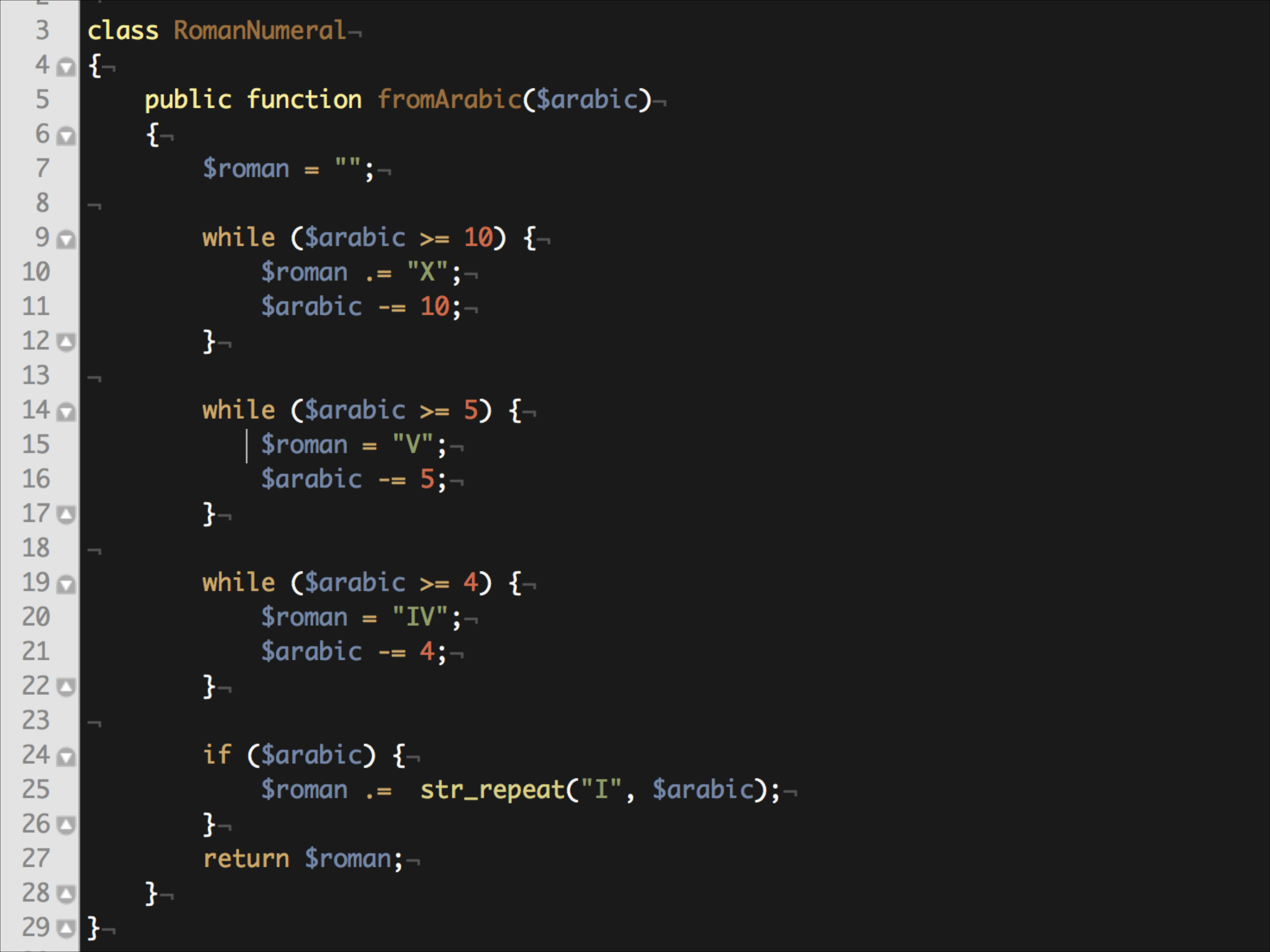
```
md@capoeira:roman_numerals $ bin/phpspec run
                         100%                          8
1 specs
8 examples (8 passed)
11ms
md@capoeira:roman_numerals $ 
```



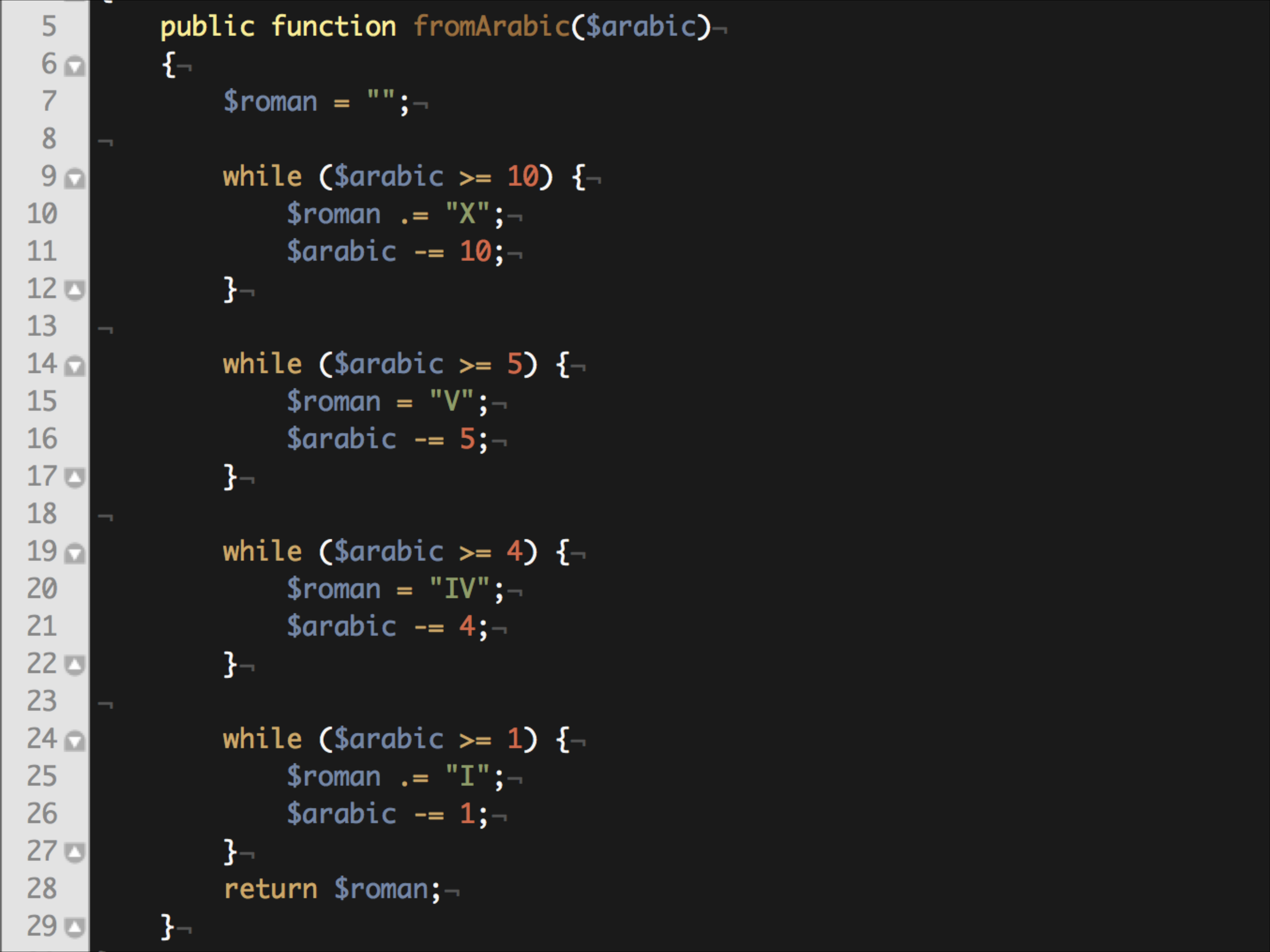
```php
class RomanNumeral
{
    public function fromArabic($arabic)
    {
        $roman = "";

        while ($arabic >= 10) {
            $roman .= "X";
            $arabic -= 10;
        }

        while ($arabic >= 5) {
            $roman = "V";
            $arabic -= 5;
        }

        while ($arabic >= 4) {
            $roman = "IV";
            $arabic -= 4;
        }

        if ($arabic) {
            $roman .=  str_repeat("I", $arabic);
        }
        return $roman;
    }
}
```



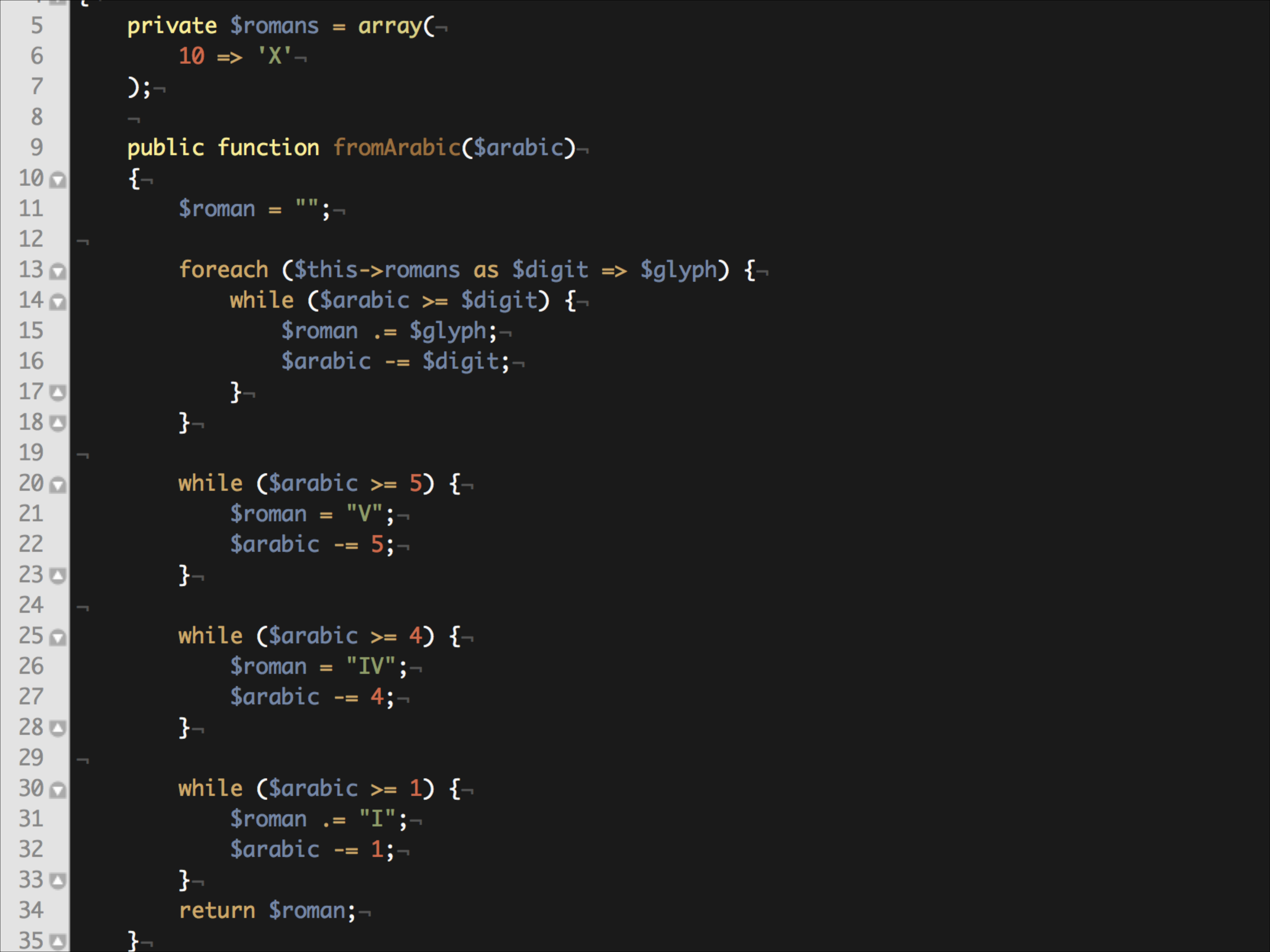
```
md@capoeira:roman_numerals $ bin/phpspec run
                      100%                        8
1 specs
8 examples (8 passed)
11ms
md@capoeira:roman_numerals $ 
```



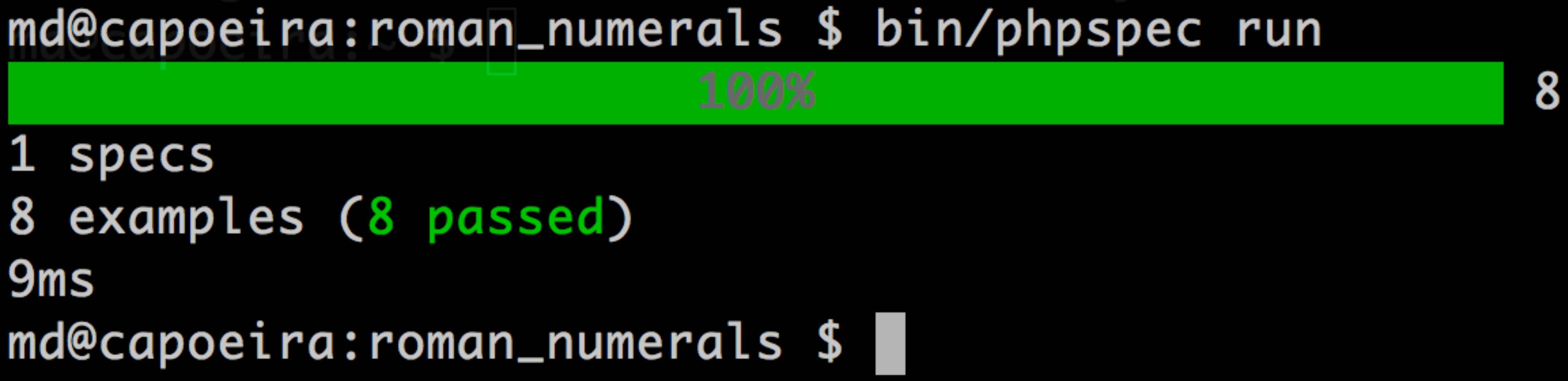
```php
class RomanNumeral
{
    public function fromArabic($arabic)
    {
        $roman = "";

        while ($arabic >= 10) {
            $roman .= "X";
            $arabic -= 10;
        }

        while ($arabic >= 5) {
            $roman = "V";
            $arabic -= 5;
        }

        while ($arabic >= 4) {
            $roman = "IV";
            $arabic -= 4;
        }

        if ($arabic) {
            $roman .=  str_repeat("I", $arabic);
        }
        return $roman;
    }
}
```



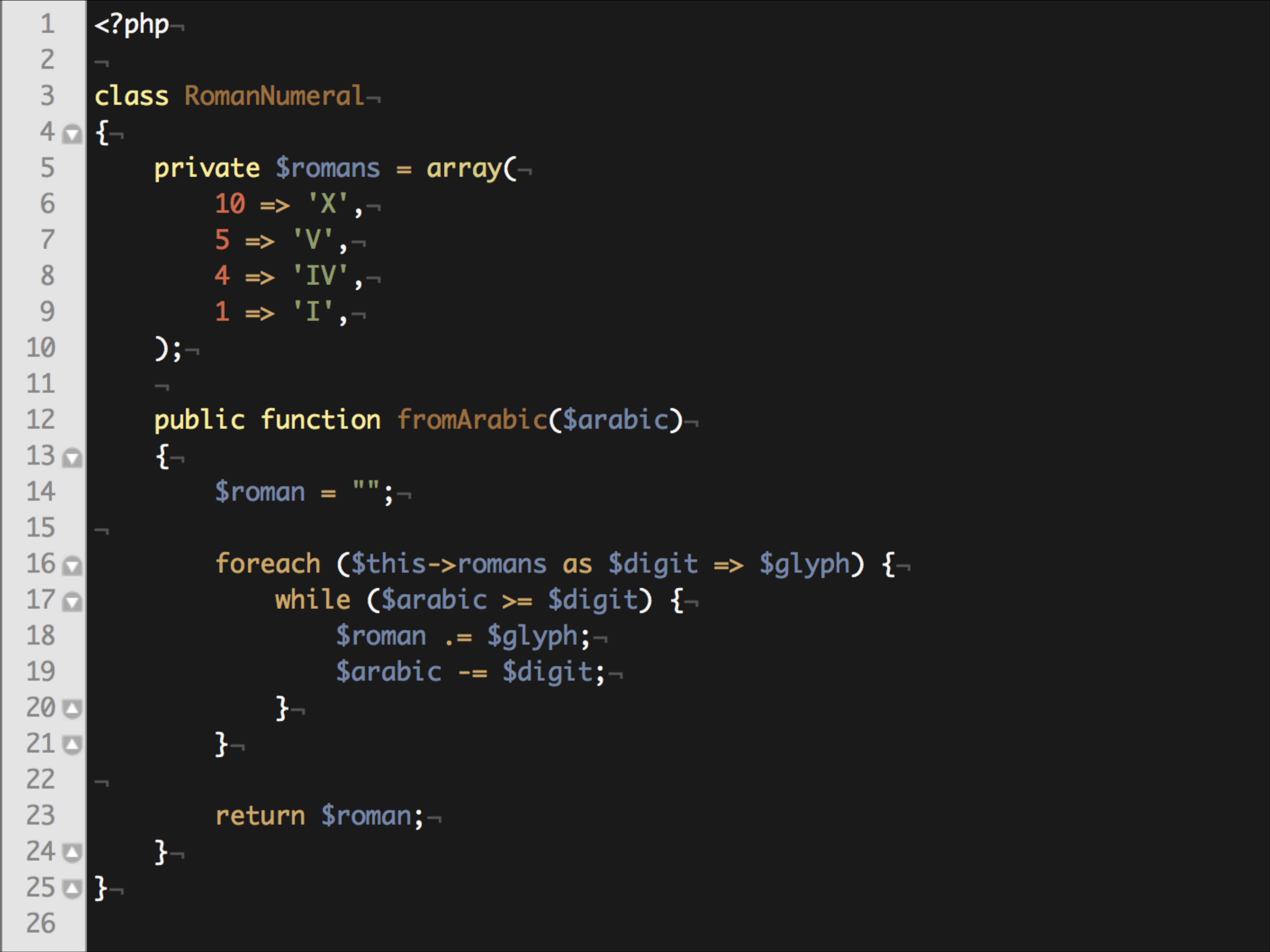
```php
    public function fromArabic($arabic)
    {
        $roman = "";

        while ($arabic >= 10) {
            $roman .= "X";
            $arabic -= 10;
        }

        while ($arabic >= 5) {
            $roman = "V";
            $arabic -= 5;
        }

        while ($arabic >= 4) {
            $roman = "IV";
            $arabic -= 4;
        }

        while ($arabic >= 1) {
            $roman .= "I";
            $arabic -= 1;
        }
        return $roman;
    }
```



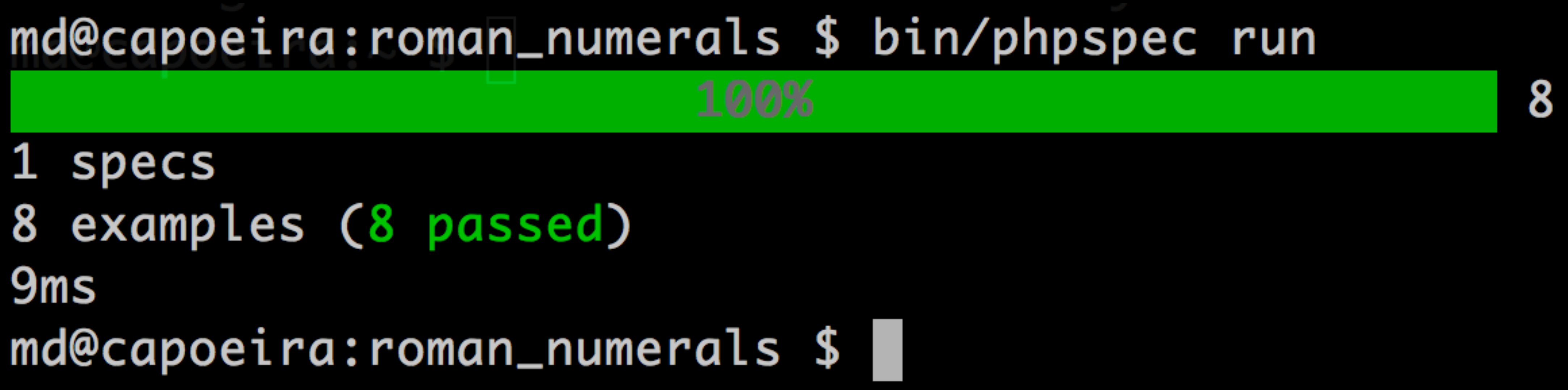
```php
    private $romans = array(
        10 => 'X'
    );

    public function fromArabic($arabic)
    {
        $roman = "";

        foreach ($this->romans as $digit => $glyph) {
            while ($arabic >= $digit) {
                $roman .= $glyph;
                $arabic -= $digit;
            }
        }

        while ($arabic >= 5) {
            $roman = "V";
            $arabic -= 5;
        }

        while ($arabic >= 4) {
            $roman = "IV";
            $arabic -= 4;
        }

        while ($arabic >= 1) {
            $roman .= "I";
            $arabic -= 1;
        }
        return $roman;
    }
```



```
md@capoeira:roman_numerals $ bin/phpspec run
                              100%                                  8
1 specs
8 examples (8 passed)
9ms
md@capoeira:roman_numerals $ 
```

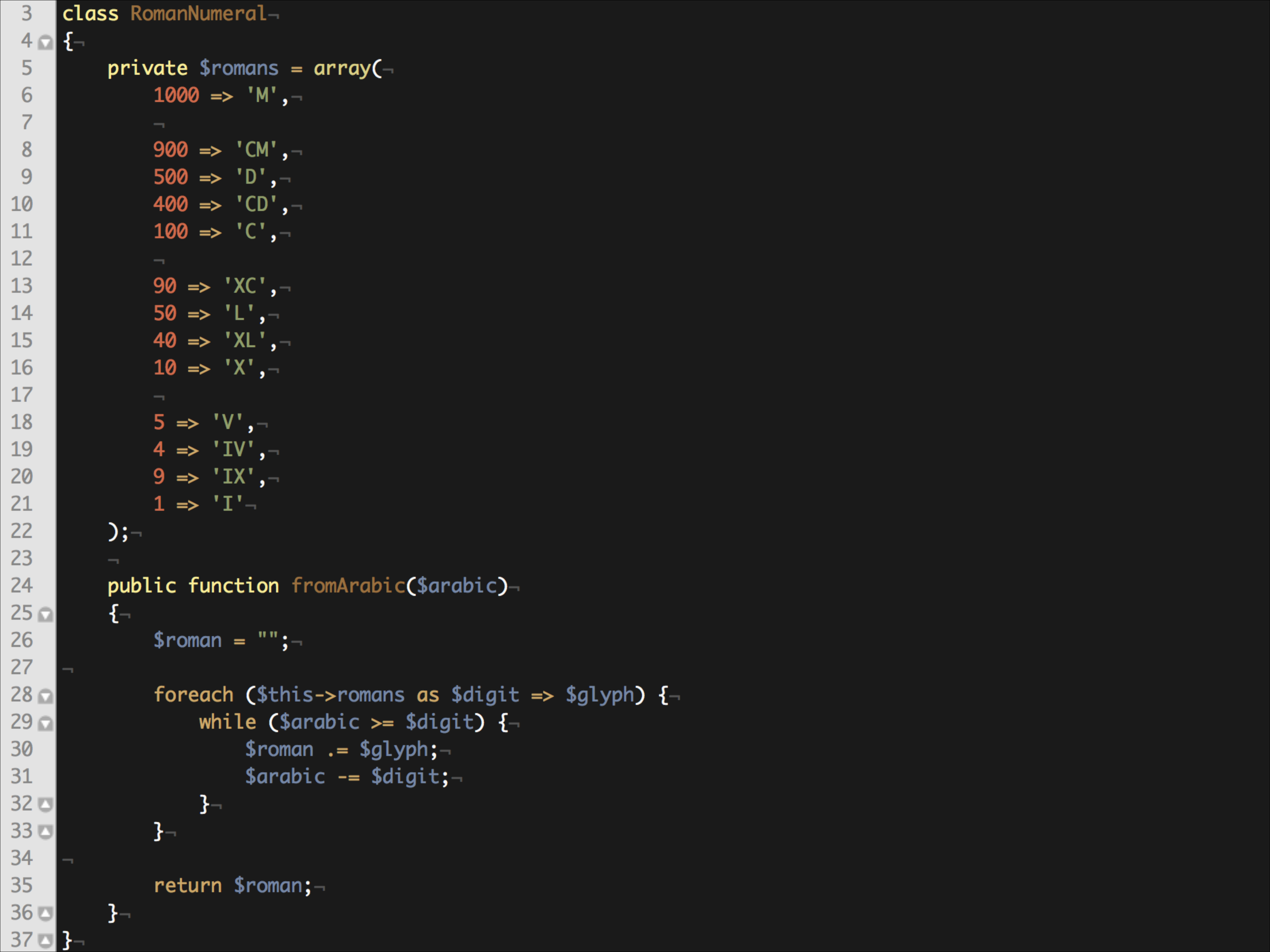


```php
<?php

class RomanNumeral
{
    private $romans = array(
        10 => 'X',
        5 => 'V',
        4 => 'IV',
        1 => 'I',
    );

    public function fromArabic($arabic)
    {
        $roman = "";

        foreach ($this->romans as $digit => $glyph) {
            while ($arabic >= $digit) {
                $roman .= $glyph;
                $arabic -= $digit;
            }
        }

        return $roman;
    }
}
```

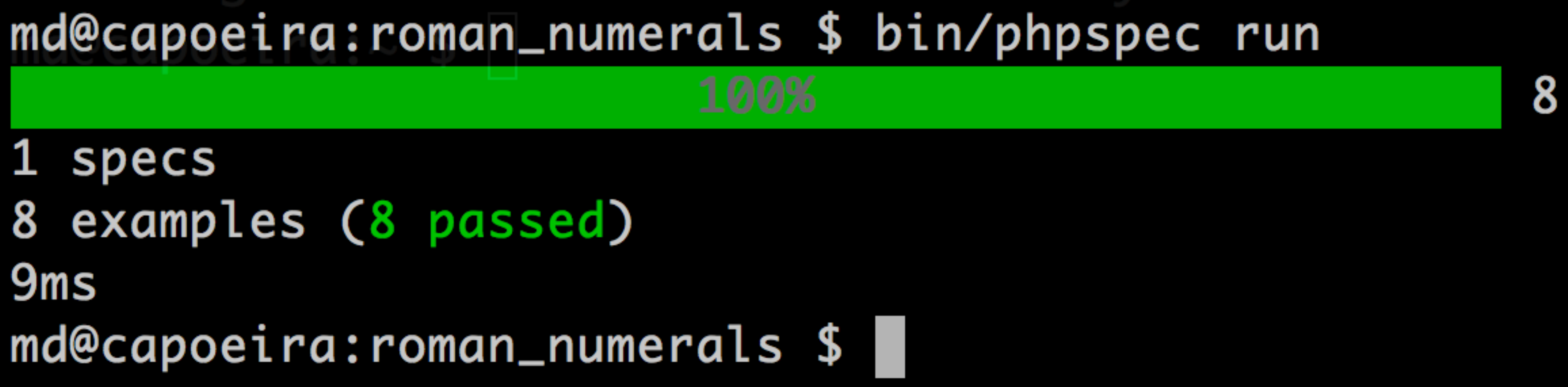


```
md@capoeira:roman_numerals $ bin/phpspec run
                       100%                          8
1 specs
8 examples (8 passed)
9ms
md@capoeira:roman_numerals $
```

```php
function it_returns_the_roman_equivalent_of_any_arabic_number()
{
    $this->fromArabic(2465)->shouldReturn("MMCDLXV");
}
```

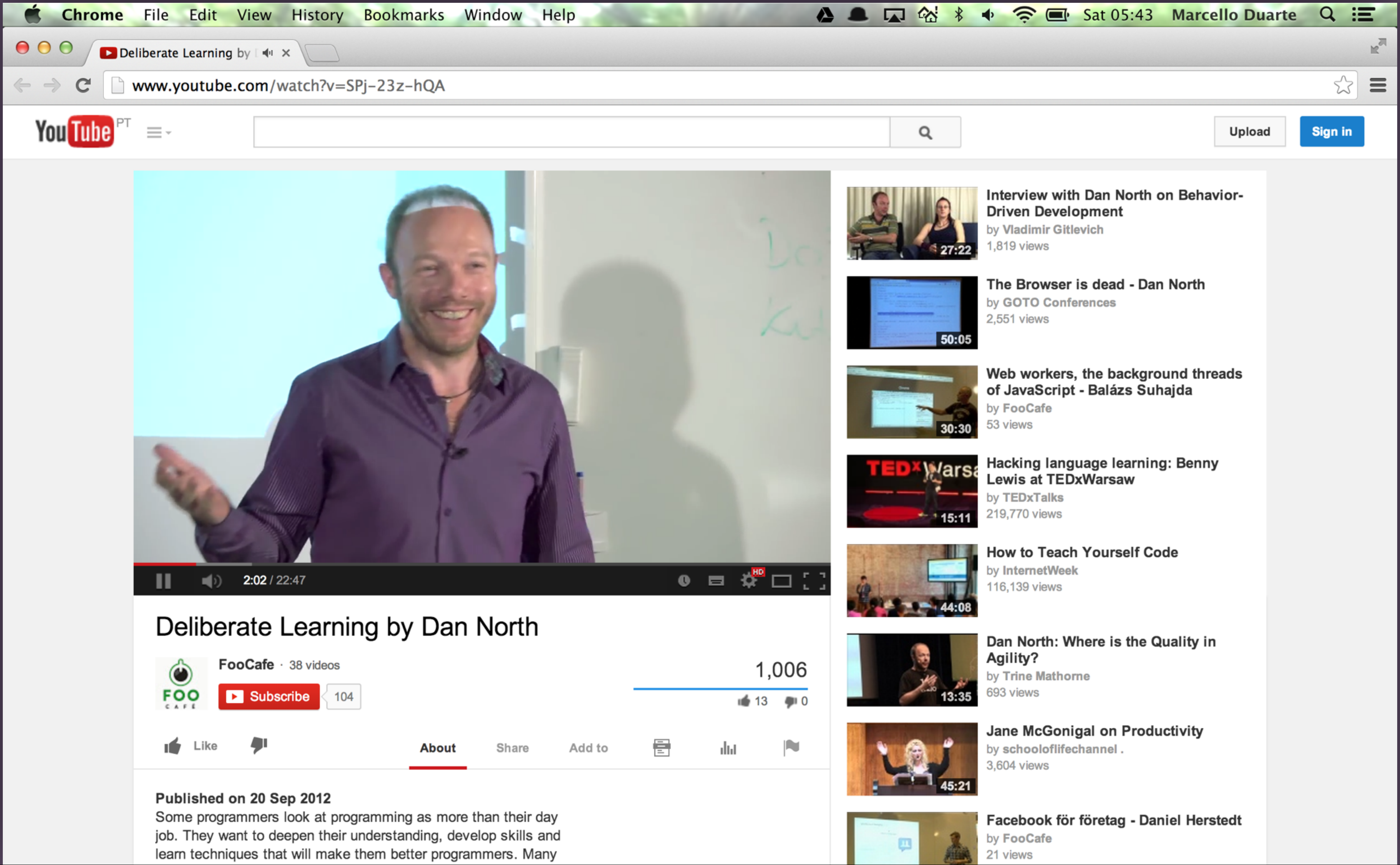


```php
class RomanNumeral
{
    private $romans = array(
        1000 => 'M',

        900 => 'CM',
        500 => 'D',
        400 => 'CD',
        100 => 'C',

        90 => 'XC',
        50 => 'L',
        40 => 'XL',
        10 => 'X',

        5 => 'V',
        4 => 'IV',
        9 => 'IX',
        1 => 'I'
    );

    public function fromArabic($arabic)
    {
        $roman = "";

        foreach ($this->romans as $digit => $glyph) {
            while ($arabic >= $digit) {
                $roman .= $glyph;
                $arabic -= $digit;
            }
        }

        return $roman;
    }
}
```



```
md@capoeira:roman_numerals $ bin/phpspec run
                              100%                                  8
1 specs
8 examples (8 passed)
9ms
md@capoeira:roman_numerals $ 
```

Katas and Learning


Chrome File Edit View History Bookmarks Window Help
Sat 05:43 Marcello Duarte
Deliberate Learning by
www.youtube.com/watch?v=SPj-23z-hQA
Upload
Sign in
2:02 / 22:47

Deliberate Learning by Dan North

FooCafe · 38 videos

Subscribe 104

1,006

13 0

Like · About · Share · Add to

Published on 20 Sep 2012

Some programmers look at programming as more than their day job. They want to deepen their understanding, develop skills and learn techniques that will make them better programmers. Many

Interview with Dan North on Behavior-Driven Development
by Vladimir Gitlevich
1,819 views
27:22

The Browser is dead - Dan North
by GOTO Conferences
2,551 views
50:05

Web workers, the background threads of JavaScript - Balázs Suhajda
by FooCafe
53 views
30:30

Hacking language learning: Benny Lewis at TEDxWarsaw
by TEDxTalks
219,770 views
15:11

How to Teach Yourself Code
by InternetWeek
116,139 views
44:08

Dan North: Where is the Quality in Agility?
by Trine Mathorne
693 views
13:35

Jane McGonigal on Productivity
by schooloflifechannel .
3,604 views
45:21

Facebook för företag - Daniel Herstedt
by FooCafe
21 views

Kata	Koan
Deliberate Practice	Deliberate Learning
• Optimising for performance	• Optimising for discovery
• Minimising variance	• Maximising variance

[North 12]

Learning TDD is learning all that TDD does for you

Mein System

Ein Lehrbuch des Schachspiels
auf ganz neuartiger Grundlage

von

A. Nimzowitsch

4. Lieferung

Berlin 1926
SCHACHVERLAG BERNHARD KAGAN :: BERLIN W 8
BEHREN-STRASSE 24

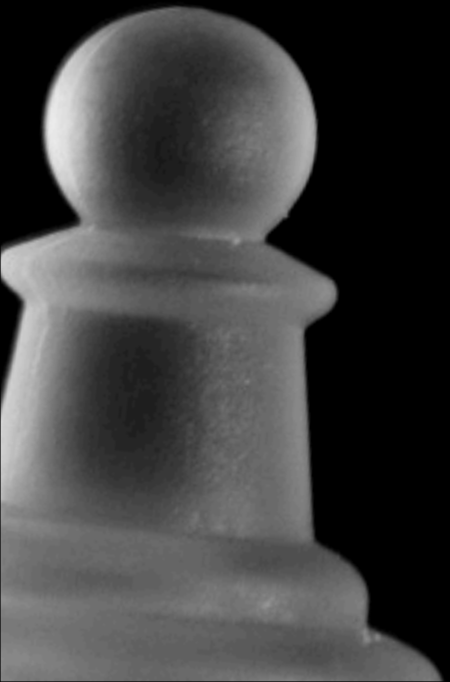
“The beauty of a move lies not in its' appearance but in the thought behind it.”

My System

elements of chess

positional play

illustrative games



First learn the elements of design
Then learn how to drive it with test

Marcello Duarte

we are hiring INVIQA

try phpspec!

follow @_md

inviqa.com/tdd-immersion-day/

Thank you !

Questions or Comments?

joind.in/10711

@_md

want to learn more? inviqa.com/tdd-immersion-day/