

Solución

Clase Marine

```
1  class Marine
2      attr_accessor :damage, :armor
3      def initialize(damage, armor)
4          @damage=damage
5          @armor= armor
6      end
7  end
8
```

Clase Marine_weapon_upgrade

```
9  class Marine_weapon_upgrade
10      attr_reader :marine
11
12      def initialize(marine)
13          @marine = marine
14      end
15
16      def damage
17          marine.damage + 1
18      end
19
20      def armor
21          marine.armor
22      end
23  end
24
```

Clase Marine_armor_upgrade

```

24
25     class Marine_armor_upgrade
26       attr_reader :marine
27
28       def initialize(marine)
29         @marine = marine
30       end
31
32       def armor
33         marine.armor + 1
34       end
35
36       def damage
37         marine.damage
38       end
39     end

```

Ejecución de Pruebas

6 kyu

PatternCraft - Decorator

☆ 61

🔗 29

👤 82% of 154

🕒 66 of 666

👤 brunolm

Ruby

▼

MRI 2.5.0

VIM

EMACS

⌵

Instructions

Output

Time: 814ms

Passed: 44

Failed: 0

Test Results:

▼ Basic tests

▶ Single upgrade (2 of 2 Assertions)

▶ Double upgrade

▶ Triple upgrade

▼ Random tests

▶ Testing for armour upgrade, starting from 3

▶ Testing for weapon upgrade, starting from 13

▶ Testing for weapon upgrade, starting from 14

▶ Testing for armour upgrade, starting from 4

▶ Testing for armour upgrade, starting from 4

Solution:

```

28     def initialize(marine)
29       @marine = marine
30     end
31
32     def armor
33       marine.armor + 1
34     end
35
36     def damage
37       marine.damage
38     end

```

✓

Impressive! You may take your time to refactor/comment your solution. Submit when ready.

✕

Sample Tests:

```

1 Test.describe('Basic tests') do
2   Test.it('Single upgrade') do
3     marine = Marine.new(10, 1)
4     Test.assert_equals(Marine_weapon_upgrade.new(marine).damage, 11)
5     Test.assert_equals(Marine_weapon_upgrade.new(marine).damage, 11)
6   end
7 end

```

RESET

TEST

SUBMIT

