

OOP i TDD w Matlabie

Wojciech Decker

24 kwietnia 2017

Aplikacja do rozpoznawania emocji w sygnale mowy

dr inż. Andrzej Majkowski

Jak wygląda większość programów w Matlabie?

- 1 skrypt
- 2 wszystko w jednym pliku
- 3 zmienne globalne
- 4 programowanie imperatywne

Co z tego wynika?

- ❶ parametry wymieszane z logiką
- ❷ kopiowanie kodu
- ❸ kod jest szyty na miarę konkretnego problemu, nie można go wykorzystać gdzie indziej
- ❹ nietestowalny

- 1 klasy (metody i pola)
- 2 analogia do rzeczywistości
- 3 SOLID
- 4 poziomy abstrakcji
- 5 warstwy

Matlab obiektowo

```
classdef FeatureMap
    properties (Access=private)
        map;
    end

    methods
        function obj = FeatureMap
            obj.map = containers.Map('KeyType','char','ValueType','double');
        end
        function val = get(obj, key)
            val = obj.map(key);
        end
        function [] = put(obj, key, value)
            obj.map(key) = value;
        end
    end
end

end
```

- 1 testy jednostkowe
- 2 przygotowanie danych, wykonanie akcji, sprawdzenie rezultatów
- 3 powtarzalność
- 4 automatyzacja
- 5 specyfikacja

Przykład testu jednostkowego

```
classdef SignalToolTest < matlab.unittest.TestCase
    properties
        tool;
    end
    methods(TestMethodSetup)
        function setUp(tc)
            tc.tool = business.SignalTool;
        end
    end
    methods(Test)
        function testSplitTo3(tc)
            sig = model.Signal([1 2 3 4 5], 1);
            actual = tc.tool.split(sig, 3000);
            tc.assertEqual(actual.data, [1 2 3; 4 5 0]);
        end
    end
end
```