

Metodika testovania

Tím:	#1, LEGENDRONE
Vedúci tímu:	Ing. Viktor Šulák
Členovia tímu:	Tomáš Čičman, Dušan Drábik, Dušan Hatváni, Patrik Kadlečík, Mário Keszeli, Martin Mocko, Lukáš Visokai
Akademický rok:	2016/2017
Autor:	Tomáš Čičman
Verzia číslo:	1
Dátum poslednej zmeny:	13.11.2016

Obsah

1. Úvod	3
2. Súvisiace metodiky a manuály	4
3. Súvisiace metodiky a manuály	5
3.1. Testovanie kódu	5
3.2. Vizuálne testovanie	6

1. Úvod

Cieľom tejto metodiky je ukázať akým spôsobom bude prebiehať testovanie projektu, spolu s konvenciami, ktoré by mali byť pri testovaní dodržané a použité. V tejto metodike sa uvádza kedy, a akým spôsobom testovať daný projekt.

2. Súvisiace metodiky a manuály

Metodika testovania sa opiera a odkazuje na tieto metodiky:

- Metodika reportovania chýb
- Guide: Writing Testable Code

3. Súvisiace metodiky a manuály

Testovanie produktu bude vykonávané na dvoch úrovniach. Týmito úrovňami je úroveň kódu a vizuálna úroveň celkového riešenia.

3.1. Testovanie kódu

Pre testovanie softvéru z pohľadu kódu budú využité unit testy, ktoré bude developer vytvárať pre vlastný kód. Tieto testy budú dodržiavať nasledovné konvencie:

- budú vytvárané mimo hlavného projektu, konkrétne v projekte s názvom DronSimTests. Tento projekt má referenciu na hlavný projekt DronSim.
- budú vytvárané vo vedľajšom projekte v triede, ktorej názov bude pozostávať z názvu pôvodnej triedy z hlavného projektu, spolu s príponou Tests (napr. EnvironmentTests.cs) Každá z týchto tried bude mať pred svojou definíciou atribút [TestClass] ako to zobrazuje obrázok č. 1.
- budú vytvárané pre jednotlivé metódy, pričom testovacie metódy budú mať názov pôvodnej metódy spolu s príponou Test (napr. InicializeTest) Pred definíciou každej metódy bude uvedený atribút [TestMethod] ako to zobrazuje obrázok č. 1.

```

namespace DronSimTests
{
    [TestClass]
    0 references | 0 changes | 0 authors, 0 changes
    public class EnvironmentTests
    {
        [TestMethod]
        0 references | 0 changes | 0 authors, 0 changes
        public void TestMethod1()
        {
        }
    }
}

```

Obrázok 1 Vzor testovacej triedy

Pri písaní testu sa bude dodržiavať tzv. AAA (Arrange, Act, Assert) vzor, ktorý pozostáva z použitia nasledujúcich sekcií:

- Arrange - v tejto sekcií sa inicializujú objekty a nastavujú jednotlivé hodnoty dát, ktoré sú následne poskytnuté metóde v teste
- Act - táto sekcia pozostáva zo samotného volania testovacej metódy s parametrami z predchádzajúcej sekcie
- Assert - na tomto mieste sa uskutočňuje verifikácia, či výstup testovacej metódy zodpovedá očakávaniu

Príklad takejto metódy je zobrazený na obrázku č. 2.

```
[TestMethod]
public void Withdraw_ValidAmount_ChangesBalance()
{
    // arrange
    double currentBalance = 10.0;
    double withdrawal = 1.0;
    double expected = 9.0;
    var account = new CheckingAccount("JohnDoe", currentBalance);
    // act
    account.Withdraw(withdrawal);
    double actual = account.Balance;
    // assert
    Assert.AreEqual(expected, actual);
}
```

Obrázok 2 Vzor test metódy

Pre testovanie je potrebné pamätať na vhodnú implementáciu kódu pre jednoduché testovanie. Pre zrýchlenie testovania a zlepšenie jeho efektívnosti je potrebné odbremeniť konštruktor vytváranej triedy od vytvárania zbytočných objektov. Z tohto dôvodu budeme implementovať tzv. *dummy* triedy, ktoré budú obsahovať implementáciu iba metódy a dáta výhradne potrebné pre test.

3.2. Vizuálne testovanie

Na základe požiadaviek produkt ownera bude prebiehať testovanie produktu z pohľadu používateľa. Pri tomto testovaní sa bude klásť dôraz na všetky požiadavky zadané product ownerom, ako napríklad plynulosť simulácie, prekryvanie objektov, atď.