Tehnici Avansate de Programare

Test-Driven Development și Recapitulare

Petru Rebeja, Marius Apetrii

15 Aprilie 2021

Facultatea de Matematică
Universitatea Alexandru Ioan Cuza, Iași

Introducere

Recapitulare

- Unit of Work — un șablon care ne permite să executăm toate modificările aferente bazei de date într-o singură tranzacție.
- Dependency Injection — o modalitate de a-i da unei instanțe variabilele de care aceasta are nevoie separând astfel crearea de instanțe de utilizarea lor.

Agenda

- `Test-Driven Development`
- Recapitulare

Test-Driven Development

Definiție

Test-Driven Development[1]
Este un stil de dezvoltare a aplicațiilor constând în trecerea frecventă prin următoarele trei activități:

- Scrierea de cod
- Scrierea de teste unitare și
- Refactorizare.

[1] https://www.agilealliance.org/glossary/tdd

Ciclul de dezvoltare TDD[2]

- Scierea unui singur test unitar menit să descrie un anumit aspect al metodei,
- Execuția testului care eșuează deoarece aspectul încă nu este implementat,
- Scrierea cantității de cod necesare pentru a trece testul,
- Refactorizarea codului scris pentru a se încadra în diverse criterii (simplitate, lizibilitate etc.).

[2] https://www.agilealliance.org/glossary/tdd

Ciclul de dezvoltare TDD

Procesul se repetă de mai multe ori ducând la acumularea de teste unitare pentru fiecare aspect al aplicației.

Avantaje

- Reduce timpul petrecut pentru depanare,
- Reduce numărul defectelor din aplicație,
- Scade gradul de acuplare și crește coeziunea.

Dezavantaje

- Crește timpul necesar pentru dezvoltare,
- Crește efortul necesar întreținerii,
- Necesită o perioadă de adaptare.

Teste unitare

- Metode care verifică un anumit aspect al modulului pe care îl testează.
- Sunt decorate cu un anumit atribut (ex.: `[TestMethod]`).
- Clasa care le conține poate fi și ea decorată cu un anumit atribut (ex.: `[TestClass]`).

Structura unui test

Un test este structurat conform șablonului `AAA`:

- Arrange — pregătirile necesare pentru a apela metoda testată: crearea de instanțe false, instanțierea clasei etc.
- Act — apelarea propriu-zisă a metodei testate și preluarea rezultatelor.
- Assert — verificarea rezultatelor și/sau comportamentului metodei testate.

Demonstrații

Încheiere

Recapitulare — TDD

- Test-Driven Development — un stil de dezvoltare software în care mai întâi se scriu testele pentru un anumit aspect iar mai apoi implementarea propriu-zisă.

Recapituralre — SOLID

S Single Responsibility Principle

O Open-Closed Principle

L Liskov Substitution Principle

I Interface Segregation Principle

D Dependency Inversion Principle

Recapitulare — `extension methods`

- Metode statice care extind funcționalitatea unui tip existent fără să-l modifice.

Recapitulare — `Repository` & `UnitOfWork`

- Repository Pattern — șablon de proiectare ce decuplează logica aplicației de accesul la date.
- Unit of Work — un șablon care ne permite să executăm toate modificările aferente bazei de date într-o singură tranzacție.

Vă mulțumesc

Succes la evaluare!