המכללה האקדמית להנדסה ירושלים

הנדסת תוכנה

8. בדיקות יחידה II

Mock / Dummy Objects

Pragmatic Programmer Tip :

Test Early. Test Often. Test Automatically.

Tests that run with every build are much more effective than test plans that sit on a shelf.

מה היום?

- ראינו: בדיקות -> בדיקות יחידה, Test Driven Development
- תזכורת
- תלות במערכת חיצונית Mock Objects, ווב, net.
- הדגמה
- הרצאה 3\תרגיל: סקר בדיקות

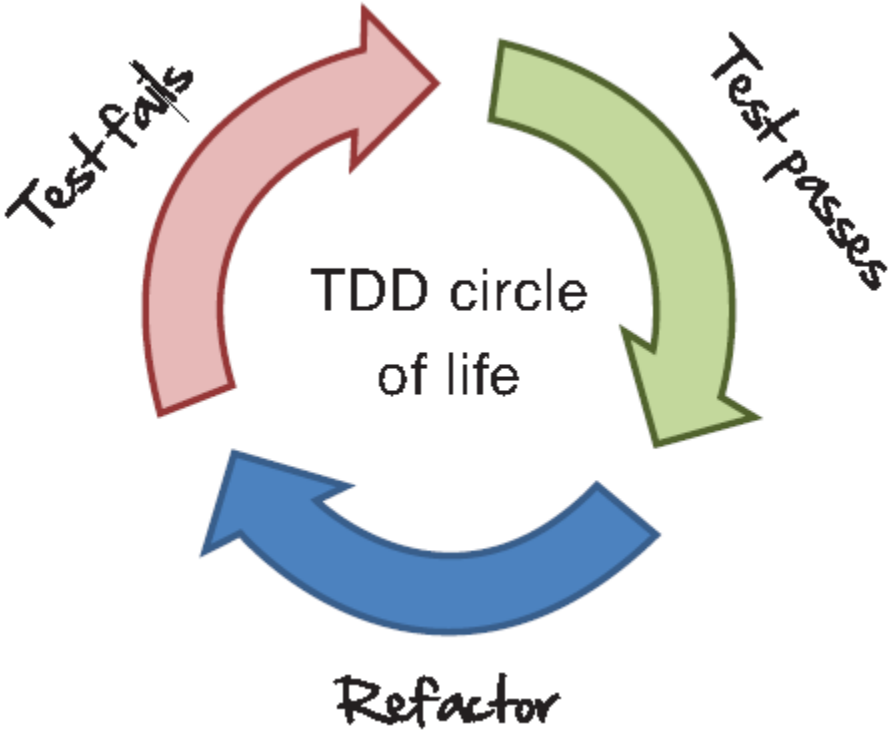
קישורים

- Using Mock Objects chapter of Pragmatic Unit Testing
- in Java with JUnit
- Fake It Til You Make It: Unit Testing Patterns With Mocks and Fakes http://www.testingtv.com/2012/11/07/unit-testing-patterns-with-mocks-and-fakes/
- Pluralsight, unit testing MVC (faking the db)
- Parameterized/White box automated unit testing: www.pexforfun.com
- Responsibility Driven Design with Mock Objects, Method&Tools, 2009
  - CRC, TDD and Java mock example

תזכורת: בדיקת יחידה טובה

- בדיקת יחידה היא קוד שקורא לקוד אחר ובודק אח"כ נכונות של טענות מסוימות על ההתנהגות הלוגית של מתודה או מחלקה.
- בדיקת יחידה תכתב בד"כ באמצעות framework
- קצרה ומורצת בקלות
- ניתנת לאוטומציה, אמינה, קריאה וקלה לתחזוקה RTFM / FIRST

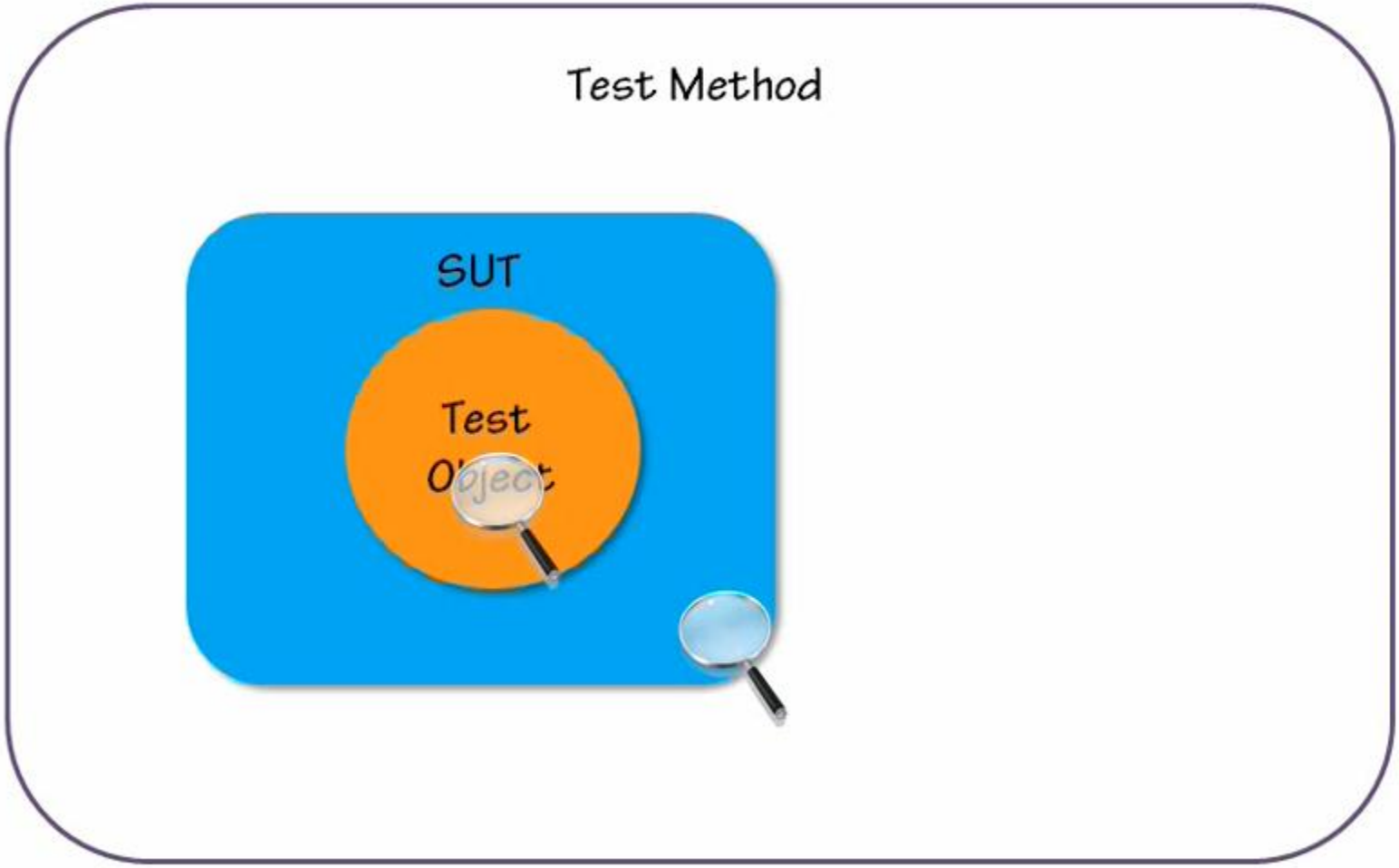
תזכורת: TDD Cycle



איך בודקים כשיש תלות בגורמים חצוניים?

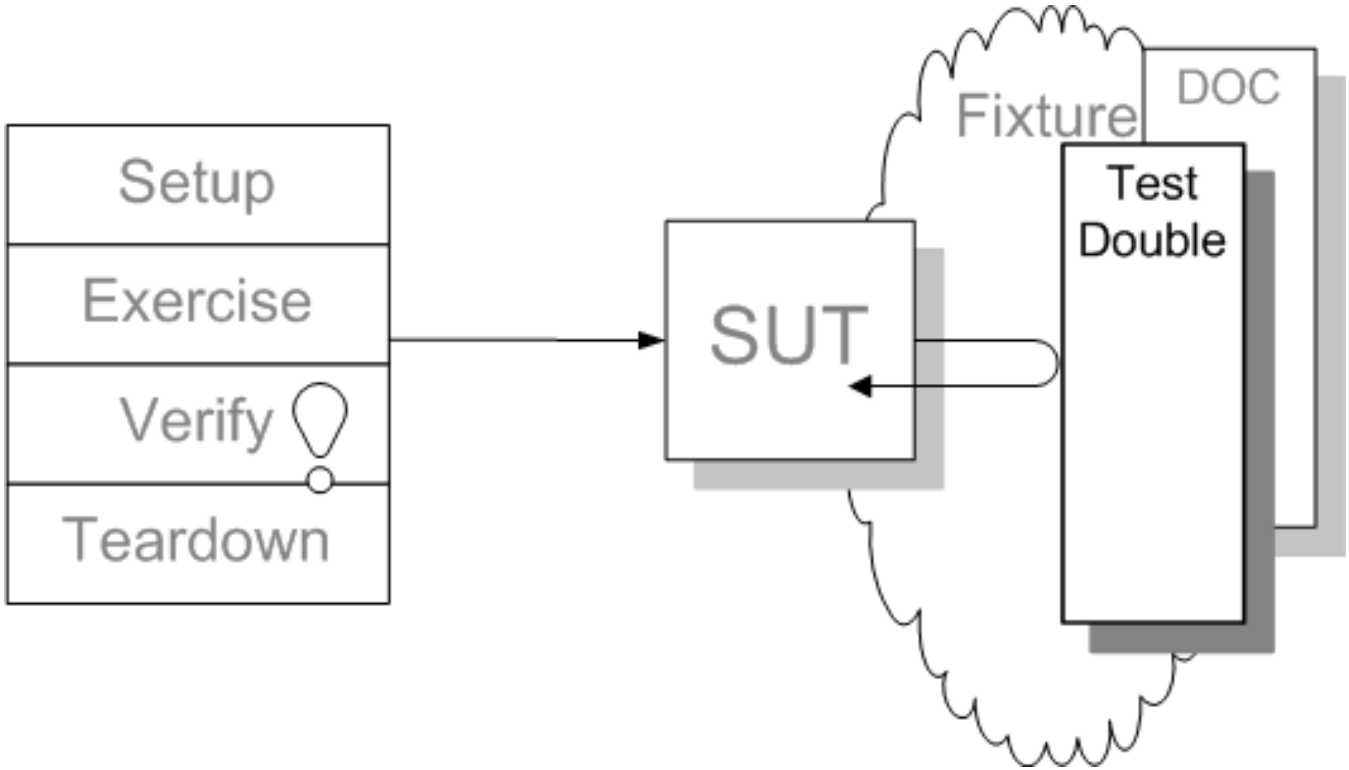
- מחלקות אחרות (שעוד לא קיימות - BDD)
- גורמים חיצוניים (למשל File System, Database) איטיים, לא עקביים

Test Isolation



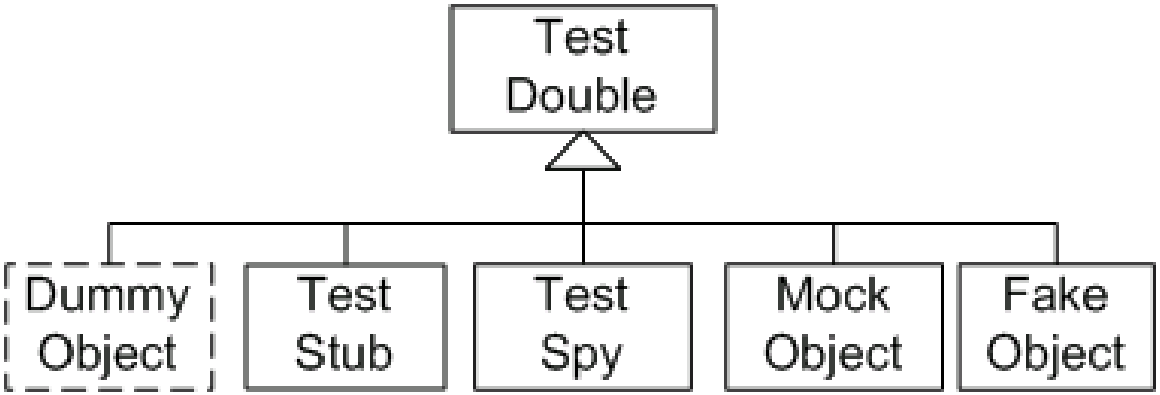
הדגמה עד הצורך...

Test Doubles



Gerard Meszaros (xunitpatterns.com)

- Test Doubles – שם כללי לאובייקטים שמחליפים אובייקטים אמיתיים, לצרכי בדיקה



Dummy

```
var list = new List<Person> {
   new Person {Name = "Sara"},
   new Person {Name = "Avi"}};
Assert.Greater(list.Count, 1);
```

Pictures from Pluralsight course TDD I

Stub

```
public class StubRepo : IOwnerRepository
{
    public IOwner FindById(int id){}

    public IOwner Save(IOwner owner)
    {
        return new Owner();
    }

    public void Delete(IOwner owner){}
}
```

Fake

```
public class FakeRepo : IOwnerRepository
{
    IList<IOwner> _owners = new List<IOwner>();
    int _idCounter = 0;

    public IOwner Save(IOwner owner)
    {
        owner.Id = _idCounter++;
        _owners.Add(owner);
        return owner;
    }

    public void Delete(IOwner owner)
    {
        var ownerToDelete = _owners.FirstOrDefault(o => o.Id == owner.Id);
        _owners.Remove(ownerToDelete);
    }
}
```

Spy

```
public class SpyDefaultView : IDefaultView
{
    public SpyDefaultView()
    {
        ShowWasCalled = false;
    }

    public void Show(DefaultVM model)
    {
        ShowWasCalled = true;
    }

    public void ShowError(string err

    public void Redirect(string url){}

    public bool ShowWasCalled { get; set; }
}
```

```
Assert.IsTrue(spy.ShowWasCalled);
```

Mock Object (אובייקט מדומה)

- אוביקט הנוצר ע"י ספריה, ניתן לקנפג את האוביקט להחזיר ערכים על פעולות, לוודא שפעולות מסוימות נקראו ועוד.
- בד"כ נרצה להשתמש בספריות, לדוגמא:

Java: mockito, jMock, EasyMock,
.Net: Nmock, moq, RhinoMock, Isolator, Nsubstitute, FakeItEasy, NUnit ...

- בד"כ יכולות לשמש ליצירת Test Doubles
- (עוד בתיכון מונחה עצמים)

מהי המטרה של mock objects?

1. לבדוק אם האובייקט הנבדק מתקשר נכון עם סביבתו
2. לספק סביבה מתאימה לבדיקת אובייקט בבדיקות יחידה
3. להגיע לכיסוי קוד גבוה
4. לאפשר לבדוק גם כשתלויות עדיין חסרות

Unit Testing a .net Web App

- Tools (&methods):
  - MS Visual Studio (IDE - free@Dreamspark, Web/UI testing) + Resharper (productivity, test runner - jce license)
  - Asp.net mvc (web framework), scaffolding
  - VS Add-ins: git provider (vcs), MSTest/NUnit (unit testing), NSubstitute/typemock (mock library), Ncrunch (coverage, continuous testing), nuget (package mgmt.)
- Patterns, Principles, Practices:
  - MVC, Repository, SOA, TDD, DRY (views)

PowerTodo Steps

- Unit test (after) main view
- Simple model test (nunit)
- Test controller-model logic & interaction
  - Scaffold controller
  - Against db
- Unit test main logic
  - Mock Repository
  - Mock DB itself (typemock isolator)
- UI Testing
- External service, e.g. Facebook
- Snippets: https://gist.github.com/4361873

Java Unit Testing

- Eclipse (IDE+test ruuner), Egit (Version Control)
- JUnit 4 (unit testing), Mockito (mocking framework)
  - add both to classpath
- TDD Example
- Mocking: Mockito.LoginServiceExample
- (git commit/push)

Mockito_LoginServiceExample.htm

נושאים נוספים

- מאפיינים שונים של xUnit (אתחולים, חריגות, ...)
- אינטגרציה\ממשק משתמש
- פרמטרים
- כיסוי
- אוטומציה \ Continuous Integration
- בדיקות לקוד קיים (Legacy Code)
- קוד מובייל \ ענן \ ווב – למשל QUnit
- כיצד להטמיע TDD בארגון?
- עוד בקורס בדיקות תוכנה (אינטל)

בפעם הבאה

- **עקרונות תיכון מונחה עצמים**
- מצגת סבב 2
  - תכנון סבב 3
- קריאה:

"Separation of Concern vs Single Responsibility Principle (SoC vs SRP)"

שאלה: מהו ההבדל העיקרי בין שני העקרונות המוזכרים?

סיכום

- אז למה בדיקות עכשיו?
- הקשר לתיכון
  - עוד בקורס המשך
- לוקח זמן עד שמקבלים רווח
  - תרגול ולימוד (<->) מתמשכים
  - Code retreats...