



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
- תלות במערכת חיצונית Mock Objects, ווב, .net
- הדגמה
- הרצאה 3\תרגיל: סקר סבב – הדגמה+מצב משימות+תכנון (רטרוספקטיבה – עצמאי)

# מקורות

- Osherove, "Interaction testing with mock objects" The art of unit testing, '09
- Meszaros, xUnit Test Patterns: Refactoring Test Code, '07
- [Mock Roles, not Objects](#), '04
- Fowler, [Mocks Aren't Stubs](#)

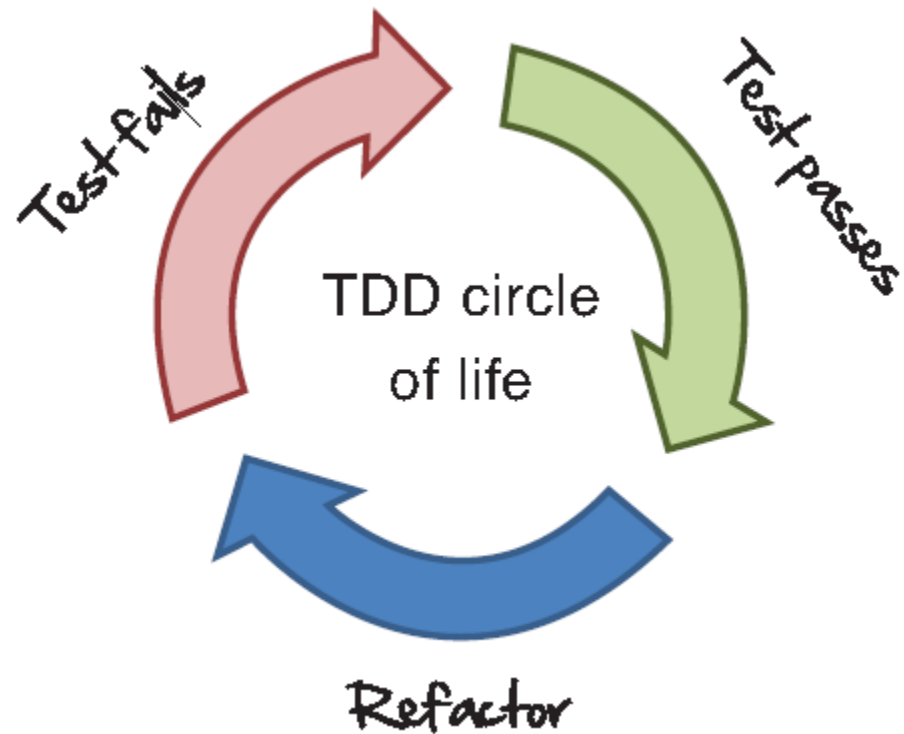
# קישורים

- [Using Mock Objects](#) chapter of Pragmatic Unit Testing
- in Java with JUnit
- The Art of Mocking  
<http://www.methodsandtools.com/archive/archive.php?id=122>
- Fake It Till 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](http://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
- FIRST

# תזכורת: TDD Cycle

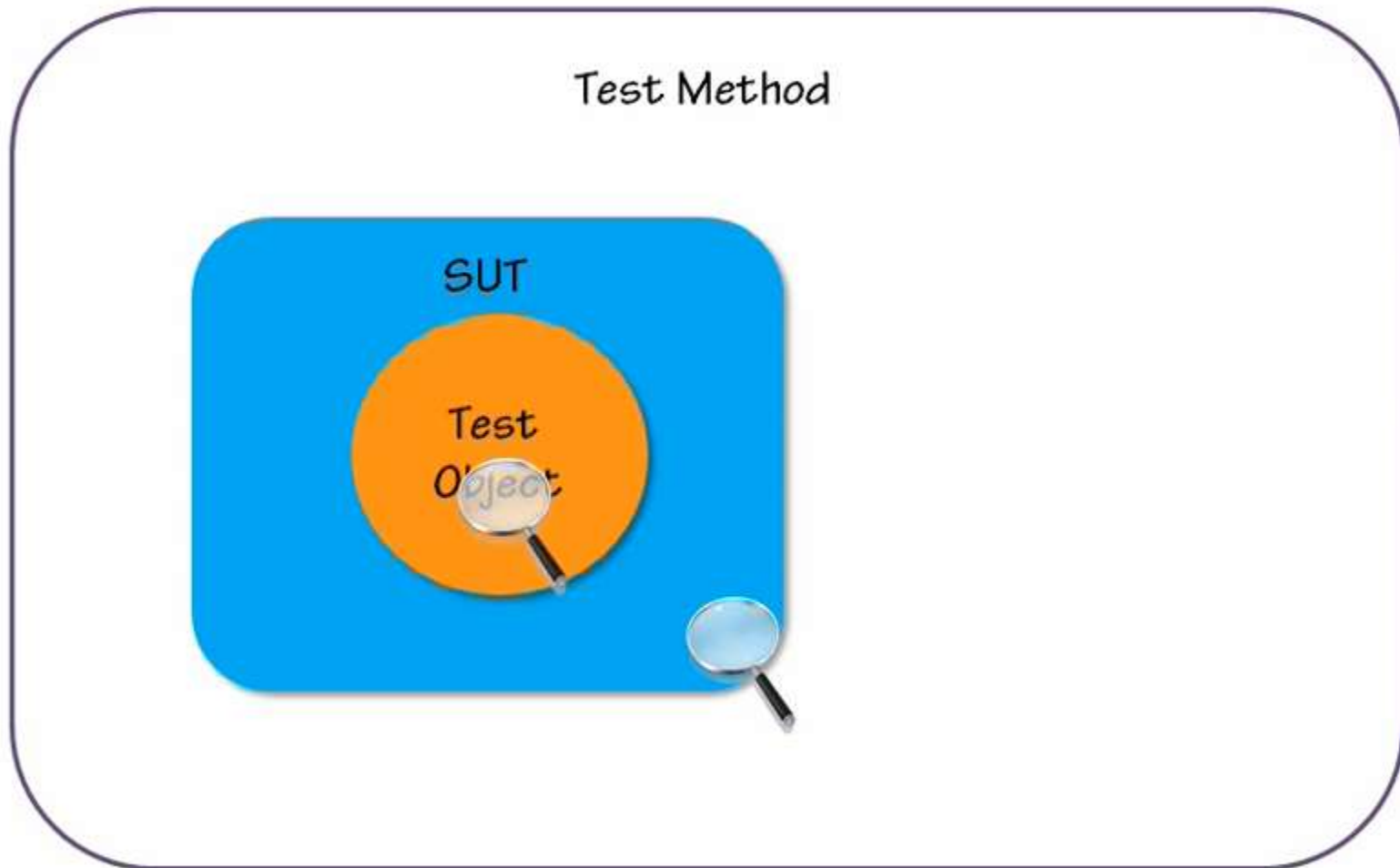


# איך בודקים כשיש תלות בגורמים חצוניים?

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



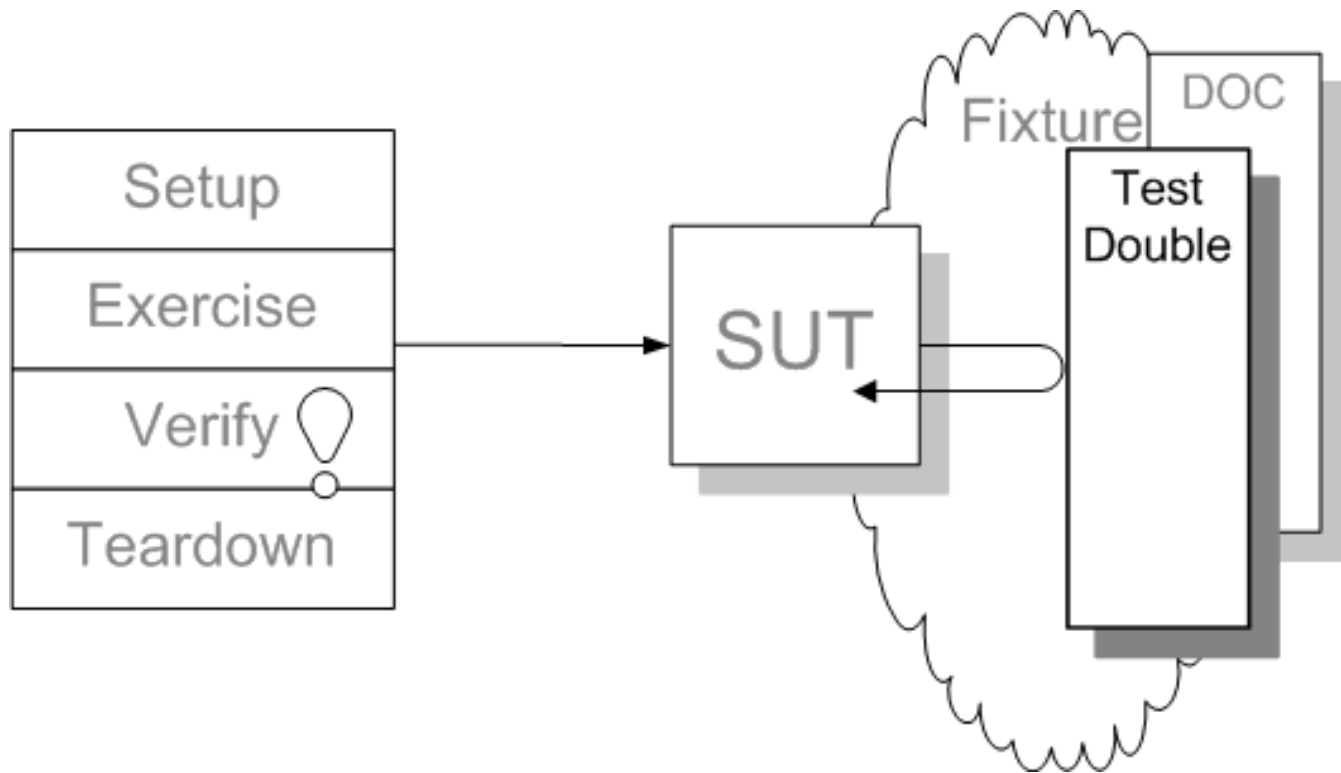
# Test Isolation





# הדגמה \ דיון

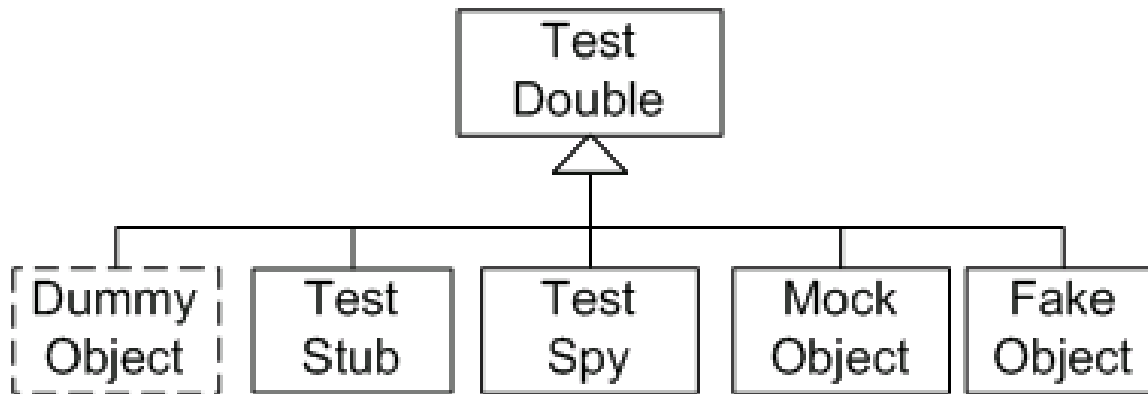
# Test Doubles



# Gerard Meszaros

([xunitpatterns.com](http://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);
```



# 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)

- אובייקט הנוצר ע"י ספריה, ניתן לקנפג את האובייקט להחזיר ערכים על פעולות, **לודא** **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 שונים
- (עוד בתיכון מונחה עצמים)



# מה אינה מטרה של mock objects?

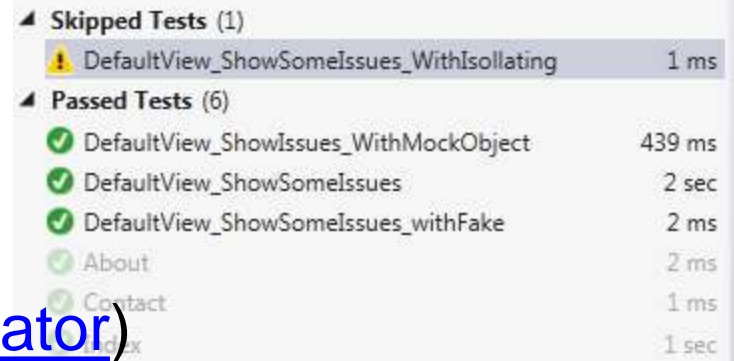
1. לבדוק אם האובייקט הנבדק מתקשר נכון עם סביבתו
2. להריץ את התלויות של אובייקט באופן אוטומטי
3. להגיע לכיסוי קוד גבוה
4. לאפשר לבדוק גם כשתלויות עדיין חסרות

# 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 (x64?)/NUnit (unit testing), [NSubstitute](#)/FakeItEasy/typemock (mock library), Ncrunch/TestView (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>



A screenshot of a test runner interface showing a list of tests. The 'Skipped Tests' section contains one test: 'DefaultView\_ShowSomeIssues\_WithIsollating' (1 ms). The 'Passed Tests' section contains six tests: 'DefaultView\_ShowIssues\_WithMockObject' (439 ms), 'DefaultView\_ShowSomeIssues' (2 sec), 'DefaultView\_ShowSomeIssues\_withFake' (2 ms), 'About' (2 ms), 'Contact' (1 ms), and 'Index' (1 sec). Each passed test is preceded by a green checkmark icon.

Skipped Tests (1)	
DefaultView_ShowSomeIssues_WithIsollating	1 ms
Passed Tests (6)	
DefaultView_ShowIssues_WithMockObject	439 ms
DefaultView_ShowSomeIssues	2 sec
DefaultView_ShowSomeIssues_withFake	2 ms
About	2 ms
Contact	1 ms
Index	1 sec

# Java Unit Testing

- Eclipse (IDE+test runner), 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 Unit x (אתחולים, חריגות, ...)
- אינטגרציה\ממשק משתמש
- פרמטרים
- כיסוי
- Continuous Integration \ אוטומציה
- בדיקות לקוד קיים (Legacy Code)
- קוד מובייל \ ענן \ ווב – למשל [JUnit](#)
- כיצד להטמיע TDD בארגון?
- עוד בקורס בדיקות תוכנה (אינטל)

# בשבוע הבא

- תיכון מתמשך (מבוא לתיכון מונחה עצמים)
- פרויקט – סבב 2
- סקר בדיקות (שבוע לפני סוף הסבב)
- קריאה:

Truyers, “[How to unit test and refactor legacy code?](#)”

- שאלה: כיצד שיפור תיכון מאפשר הוספת בדיקות, הדגם.

# סיכום

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