

USING TEST DOUBLES (MOCKS)

Types of test doubles

- **Dummy object** - used when a parameter is needed for the tested method but without actually needing to use the parameter.
- **Fake object** - used as a simpler implementation, e.g. using an in-memory database in the tests instead of doing real database access
- **Test stub** - used for providing the tested code with "indirect input".
- **Mock object** - used for verifying "indirect output" of the tested code, by first defining the expectations before the tested code is executed.

Šta je lažiranje objekata? (*mocking objects*)

- U OOP, lažirani objekti (*mock objects*) se koriste u simulacijama, kako bi imitirali ponašanje stvarnih objekata na kontrolisan način.
- Programer lažira objekte radi provere ponašanja, po istom principu po kojem mašinski (auto) inženjer koristi lutku da bi simulirao ponašanje čoveka pri sudaru automobila.
- Lažirani objekti mogu biti od velike pomoći pri simuliranju rada baze podataka, mrežnog (*web*) servisa, događaja koji kreira klijent, spajanja sa spoljnom mrežom, kao i hardvera na kojem se izvršava program.
- Lažirani objekti imaju i mogućnost da proizvedu razne otkaze (*failure*) koje je teško reprodukovati u stvarnom svetu, kao što su: loša veza sa mrežom, spor server, zagušenje mreže i slično.

Mocking Frameworks

- Mock object frameworks make it very easy to create test doubles, set expectations, with or without accompanying dummy values that should be returned, with or without throwing exceptions, raising events etc.
- Available frameworks:
 - NSubstitute
 - Moq
 - Rhino.Mocks
 - NUnit Mocks
 - Deprecated, not recommended

Moq

- Supports Arrange-Act-Assert testing
- All test double types covered with one entity called *mock*
 - Supports *loose* (default) and *strict* behaviors
 - Supports *partial* mocking
- To configure project, ensure that following steps are taken
 - Add reference to the **Moq.dll** assembly in unit test project
 - Version 4.0 of the library is placed on our source code management system, within “*Common/3rdPartyLibs/UnitTestFramework/NUnit/Moq/NET40*” repository
 - Add **Moq** namespace to files with test fixture class declaration

Moq methods

- **Creating** a mock

```
new Mock<T> ()
```

- *T* - interface to mock

- **Accessing** mocked interface

```
Mock<T>.Object
```

- **Setting** mock **behavior**

```
Mock<T>.Setup (Get/Set) <R> (action)
```

- **Configuring** **return values**

```
IReturns<T, R>.Returns (value/action)
```

- **Throwing** **exceptions**

```
IThrows.Throws<E> ()
```

- **Preparing** **callbacks:**

```
ICallback.Callback<T> (action)
```

Moq methods

- Matching **arguments**

`It.IsAny<P>()`

`It.Is<P>(value)`

- Specifying **expectations**

`IVerifies.Verifiable()`

- Checking **expectations**

`Mock.Verify(All)()`

`Mock<T>.Verify(Get/Set)(action)`

NSubstitute

- NSubstitute je jednostavna biblioteka za lažiranje objekata u .NET okviru.
- Posедуje jezgrovitu sintaksu, koja pomaže da se zadrži pažnja na nameni samog testa, a ne na konfigurisanju testova.
- Često korišćene funkcionalnosti su lako dostupne i jednostavne za upotrebu, a naredbe streme što više ka prirodnom, razgovornom jeziku.
- Osnovna upotreba:
 - `calculator = Substitute.For<ICalculator>();`
- URL: <http://nsubstitute.github.io/> & <https://github.com/nsubstitute/NSubstitute>



Zadatak 2

Implementirati interfejs Račun koji sadrži:

`IOsoba` Vlasnik

`string` BrojRacuna

`Decimal` Stanje

`Decimal` DozvoljeniMinus

Konstruktor treba da primi prilikom inicijalizacije objekat Vlasnik, broj računa I dozvoljen minus.

- Vlasnik, broj računa ne smeju biti null
- Dozvoljeni minus mora biti veći od 0

Realizovati funkciju ProveriBrojRacuna koja prima objekat Vlasnik I broj racuna I proverava:

- Da broj racuna sadrži 19 cifara
- Da se jmbg vlasnika računa nalazi kao podstring u broju računa počevši od 4 cifre
- Da prve tri cifre ne smeju biti 0

Napisati test slučajeve koji pokrivaju slučajeve:

- Konstruktor prima dobre parametre
- Konstruktor prima granične paramete
- Konstruktor prima loše parametre

Koristiti Mock za interfejs Osoba prilikom testiranja Računa