

## Stubs & Mocks *Online ordering*

Cilj zadatka jeste izolacija zavisnosti i pisanje testova za definisani sistem pod testom upotrebom koncepta lažnih implementacija.

- Ne postoji grafički interfejs.
- Radi se samo sa objektima i NUnit testovima.
- Nije potrebna *main* metoda.

### Opis zadatka:

Pred vama se nalazi deo pametnog sistema koji služi za online poručivanje proizvoda. Poručivanje funkcioniše tako što sistem nakon što korisnik kreira porudžbinu (eng. *order*) sa svim željenim proizvodima odrađuje niz koraka u cilju čuvanja i potvrde porudžbine. Prvo se vrši dobavljanje neophodnih korisničkih podataka, kao što su *userId* i *userEmail*. Nakon ovoga vrši se provera da li su svi proizvodi u porudžbini dostupni. Ukoliko makar jedan nije dostupan nastaje greška i porudžbina neće biti sačuvana, dok ukoliko jesu vrši se uklanjanje datih proizvoda sa lagera (kreira se novo stanje na lageru), porudžbina se čuva u bazi i korisniku koji je načinio porudžbinu se šalje mejl potvrde. Sistem takođe ima sposobnost da obradi greške koje mogu nastati tokom ove procedure.

### Struktura projekta:

Projekat se sastoji od klase *ProductOrdering* koja vrši gore opisano čuvanje porudžbine i to uz pomoć metode *PlaceOrder* koja za ulazni parametar uzima ID porudžbine koju je potrebno proveriti i sačuvati. Ovo ujedno predstavlja i sistem pod testom (*System Under Test*) za ovaj zadatak. Pored ove klase u strukturi projekta su prisutni i folderi *Interfaces* i *Util*. Folder interface sadrži sve neophodne interfejse za zadatak, dok folder *Util* sadrži sve „konkretnе“ implementacije zavisnosti koje koristi sistem pod testom.

Funkcionalnost metode *PlaceOrder* je podeljena na sledeće segmente (korake):

1. Dobavljanje neophodnih podataka o logovanom korisniku
2. Provera dostupnosti svakog proizvoda u porudžbini
3. U zavisnosti od dostupnosti proizvoda:
  - 3.1 Skidanje proizvoda sa lagera
  - 3.2 Čuvanje porudžbine u bazi
  - 3.3 Slanje mejla potvrde korisniku
  - 3.4 Kreiranje greške da nisu svi proizvodi dostupni – Makar jedan proizvod u porudžbini nije dostupan
4. Obrada grešaka (ukoliko je došlo do nekih)
  - 4.1 Ukoliko je nastala greška da neki proizvodi nisu dostupni čisti se baza i vraća se staro stanje lagera
  - 4.2 Loguje se greška pomoću klase *ErrorLogger*

Svi proizvodi su dostupni

Radi testiranja potrebno je kreirati testni projekat tipa *Class Library (.Net Framework)* pod nazivom *OnlineOrdering.UnitTests*. Testnu klasu koja testira definisani sistem pod testom imenovati *ProductOrderingTests*. Prilikom pisanja konkretnih testova držati se opisnih naziva testova.

Primer naziva testa: *PlaceOrder\_ValidInput\_Success()* ili *PlaceOrder\_ValidanInput\_UspesnoIzvršenje()*

### Testni zahtevi:

1. Testirati scenario uspešne provere i čuvanja porudžbine.
2. Testirati scenario nastanka greške prilikom provere dostupnosti proizvoda (korak 2).

3. Upotrebom *NSubstitute framework*-a testirati scenario nastanka greške prilikom slanja email-a korisniku sa porukom "*Failed to send email*".
4. Testirati scenario kada neki od proizvoda nisu dostupni.
5. Upotrebom *NSubstitute framework*-a testirati scenario nastanka greške prilikom skidanja proizvoda sa lagera.