

Osnove testiranja softvera

Inženjerstvo informacionih sistema

Fakultet tehničkih nauka

Online ordering

Cilj zadatka je da se provežbaju koncepti koji su u toku semestra obrađivani na vežbama. Krajnji rezultat završenog zadatka treba da bude jedan skup testova (testnih slučajeva) u okviru kojih će studenti iskoristiti navedene koncepte:

- Pisanje klasičnih Unit testova,
- Pisanje Data Driven Unit testova,
- Pisanje testova koji zahtevaju izolaciju zavisnosti,
- Test Driven Development i
- Korišćenje NSubstitute framework-a.

Opis zadatka:

Pred vama se nalazi deo pametnog sistema koji služi za online poručivanje proizvoda. Korisnicima sistema treba da budu na raspolaganju sledeće funkcionalnosti:

1. Kreiranje porudžbine,
2. Obrada porudžbine:
 - a. Računanje cene proizvoda iz porudžbine,
 - b. Računanje troškova prevoza,
 - c. Računanje popusta,
 - d. Računanje ukupne cene porudžbine,
3. Provera stanja korisničkog računa,
4. Uplata novčanih sredstava na račun i
5. Odobravanje kredita za finansiranje porudžbine.

Poručivanje, odnosno metoda *PlaceOrder*, funkcioniše tako što sistem obrađuje zahtev za odgovarajućom porudžbinom sa svim željenim proizvodima. Porudžbina se obrađuje i potvrđuje kroz niz koraka:

1. Dobavljanje korisničkog Id-a i
2. Dobavljanje informacije o raspoloživom stanju za ulogovanog korisnika
3. Provera dostupnosti svakog proizvoda u porudžbini
4. Ukoliko su svi proizvodi dostupni:
 - 4.1 Kalkulacija cene porudžbine
 - 4.2 Provera da li korisnik ima dovoljno sredstava na računu
 - 4.3 Ukoliko ima, vrši se:
 - i. Čuvanje porudžbine u bazi
 - ii. Smanjivanje stanja na računu korisnika za vrednost porudžbine
 - iii. Slanje mejla o potvrdi porudžbine
5. U svim ostalim slučajevima loguje se poruka o grešci.

Osnove testiranja softvera

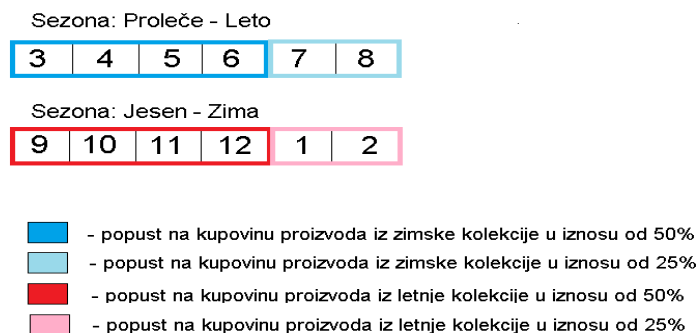
Inženjerstvo informacionih sistema

Fakultet tehničkih nauka

Svaka od prethodno navedenih funkcionalnosti u kodu treba da predstavlja pojedinačnu logiku koja će biti smeštena u zasebne metode:

1. *PlaceOrder* – metoda koja će kao ulazni parametar prihvatati porudžbinu (*Order*) koju korisnik želi da kreira. Jedna porudžbina može da obuhvati više proizvoda. Svaki od koraka za postavljanje porudžbine, gore opisanih, treba da predstavlja adekvatnu aktivnost u kodu.
2. *CalculatePrice* – metoda koja će kao ulazni parametar prihvatati čitav proizvod, količinu istog i naznaku da li korisnik želi da mu se porudžbina obračunava u dinarima ili će se izvršiti konverzija u evre. Proizvod ima određenu vrednost koja je iskazana u dinarima i u zavisnosti od oznake koja se prosleđuje kao ulazni parametar metode, sama porudžbina računa se u dinarima ili se vrši konverzija u evre. Krajnja vrednost porudžbine se dobija tako što se pomnožena vrednost proizvoda sa količinom, i u zavisnosti od potrebe za konverzijom podeli sa trenutnom vrednošću kursa dinara u evre.
3. *CalculateTransport* – metoda koja će kao ulazni parametar prihvatati iznos izračunate cene proizvoda u evrima. Ukoliko je vrednost ulaznog parametra veća od 30 evra, korisnik će biti oslobođen dodatnog troška prevoza. U suprotnom, troškovi prevoza iznose 5 evra. Metoda treba da vrati ukupan iznos porudžbine sa ukalkulisanim troškovima prevoza.
4. *CalculateDiscount* – metoda koja će kao ulazni parametar prihvatati čitavu porudžbinu (*Order*). Ukoliko je cena prosleđene porudžbine u dinarskoj valuti, korisnik će dobiti popust ako je vrednost porudžbine veća od 10000 u iznosu od 10%. Takođe, ukoliko je porudžbina data u dinarskoj valuti i broji više od 3 proizvoda, korisnik će dobiti dodatnih 7% popusta. Ukoliko je prosleđena vrednost porudžbine u evrima, korisnik dobija 12% popusta nezavisno od broja proizvoda.

Pored ovih popusta, na osnovu datuma porudžbine i kolekcije kojoj željeni proizvod koji se kupuje pripada, korisnik može ostvariti i dodatne popuste prema modelu koji je prikazan na narednoj slici (Slika 1):



NAPOMENE:

1. Broj u okviru polja predstavlja redni broj meseca.
2. Svaka sezona traje 6 meseci.
3. Popusti se odnose samo na kupovinu proizvoda iz druge sezone. Na primer, ukoliko korisnik želi da u desetom mesecu u godini (oktobru) kupi proizvode iz kolekcije Proleće - Leto, on će na cenu proizvoda ostvariti dodatnih 50% popusta.

Slika 1 - Model dobijanja popusta u zavisnosti od kolekcije

Osnove testiranja softvera

Inženjerstvo informacionih sistema

Fakultet tehničkih nauka

5. *CalculateOrder* – metoda koja kao ulazne parametre prihvata porudžbinu koja se računa, a treba da izračuna i vrati ukupnu cenu čitave porudžbine, sa uračunatim troškovima prevoza i potencijalnog popusta, pozivajući prethodno objašnjene metode u narednom redosledu:
 - a. *CalculatePrice*,
 - b. *CalculateTransport* i
 - c. *CalculateDiscount*.
6. *CheckAccountStatus* – metoda koja kao ulazni parametar prihvata korisnika sistema (*User*) i treba da vrati njegovo trenutno stanje računa.
7. *AccountPayment* – metoda koja kao ulazni parametar prihvata korisnika sistema (*User*) i iznos koji će se uplatiti na njegov lični račun. Pri uplati, treba proveriti da li korisnik ima odobren kredit na svoje ime i u slučaju da ima, smanjiti iznos kredita koji korisnik duguje. Ukoliko korisnik nema nikakvih dugovanja, odnosno kredit mu je u celosti isplaćen ili nije ni uzet na njegovo ime, novčana sredstva uplaćuju se na njegov račun tako što se povećava njegovo trenutno stanje. Ovu metodu koristiti i za smanjivanje korisničkog računa za iznos porudžbine u slučaju da je ista odobrena.
8. *CheckCreditAvailability* – metoda koja kao ulazne parametre prihvata korisnika sistema (*User*), iznos za koji korisnik želi da uzme kredit i rok na koji želi da otplaćuje kredit iskazan u mesecima. Da bi se odobrio kredit korisniku, neophodno je da se zadovolje sledeći uslovi:
 - a. Iznos za uzimanje kredita ne sme biti manji od 100.000 iskazan u dinarskoj vrednosti,
 - b. Rok otplate kredita ne sme biti veći od 24 meseca,
 - c. Korisnik ne može imati dva odobrena kredita istovremeno,

Ukoliko su sva ova pravila zadovoljena, korisniku se kredit odobrava na taj način što mu se trenutno stanje računa uvećava za željeni iznos kredita i ujedno se i uvećava njegovo kreditno zaduženje (obeležje *CreditRepayment*) za isti iznos. Pored toga, status kreditne sposobnosti korisnika (*CreditAvailable*) takođe treba da se promeni kako korisnik ne bi mogao da uzme još jedan kredit na svoje ime. Metoda treba da vrati informaciju da li je kredit odobren ili nije.

Upotrebom TDD (*Test Driven Development*) pristupa implementirati sve prethodno navedene zahteve. Zahteve je potrebno podeliti na male celine (*Slices*) i za svaku od njih napisati odgovarajuće testove i potom potrebnu logiku. Dokumentovati svaku identifikovanu celinu u redosledu po kojem su one kreirane u okviru template excel dokumenta *OnlineOrdering_template* koji se nalazi u okviru početnog repozitorijuma na vašem Github nalogu.

Osnove testiranja softvera

Inženjerstvo informacionih sistema

Fakultet tehničkih nauka

Struktura projekta:

U okviru projekta inicijalno postoji dva foldera:

1. *Interfaces* i
2. *Util*.

Folder *Interfaces* sadrži sve interfejsse koji treba da budu implementirani od strane stvarnih klasa, kao i lažnih implementacije, čime se postiže zavisnost od apstrakcije, a ne od stvarne implementacije. Samim time moguća je izolacija zavisnosti prilikom pisanja Unit testova.. Sve lažne implementacije koje studenti kreiraju, treba da budu smeštene u novi folder koji će biti nazvan *Fakes*. U okviru klasa koje će lažne implementacije koristiti, neophodno je obezbediti podmetanje istih putem konstruktora i putem property-a klase.

Sa druge strane, folder *Util* sadrži klase *Product* koja omogućava kreiranje univerzalnog proizvoda koji će biti korišćen u daljem toku zadatka, klasu *User* koja sadrži osnovne informacije o korisniku sistema i klasu *Order* koja modeluje porudžbinu u ovom sistemu.

Studenti u okviru foldera *Util*, treba da kreiraju tri klase od kojih će svaka pojedinačno predstavljati sistem pod testom (*System Under Test*):

1. *ProductOrdering* - koja će sadržati samo metodu *PlaceOrder*
2. *PriceCalculator* – koja će sadržati metode *CalculatePrice*, *CalculateTransport*, *CalculateOrder*, *CalculateDiscount*.
3. *UserPayment* – koja će sadržati samo metodu *CheckAccountStatus*, *AccountPayment* i *CheckCreditAvailability*

Radi testiranja potrebno je kreirati testni projekat tipa *Class Library (.Net Framework)* pod nazivom *OnlineOrdering.UnitTests*. Za svaku od kreiranih klasa neophodno je kreirati jednu testnu klasu u okviru testnog projekta (npr. *ProductOrderingTest* za klasu *ProductOrdering*). Prilikom pisanja konkretnih testova držati se opisnih naziva testova.

Primer naziva testa: *PlaceOrder_ValidInput_Success()* ili *PlaceOrder_ValidanInput_UspesnoIzvršenje()*.