

Zadaća br. 2

Izvještaj o inspekciji koda

Informacije o timu

Dodijeljeno programsko rješenje: [Grupa 6] Kupid

Ime i prezime: Sara Makešoska-Džebo

Broj indexa: 17814

Ime i prezime: Medina Kapo

Broj indexa: 18076

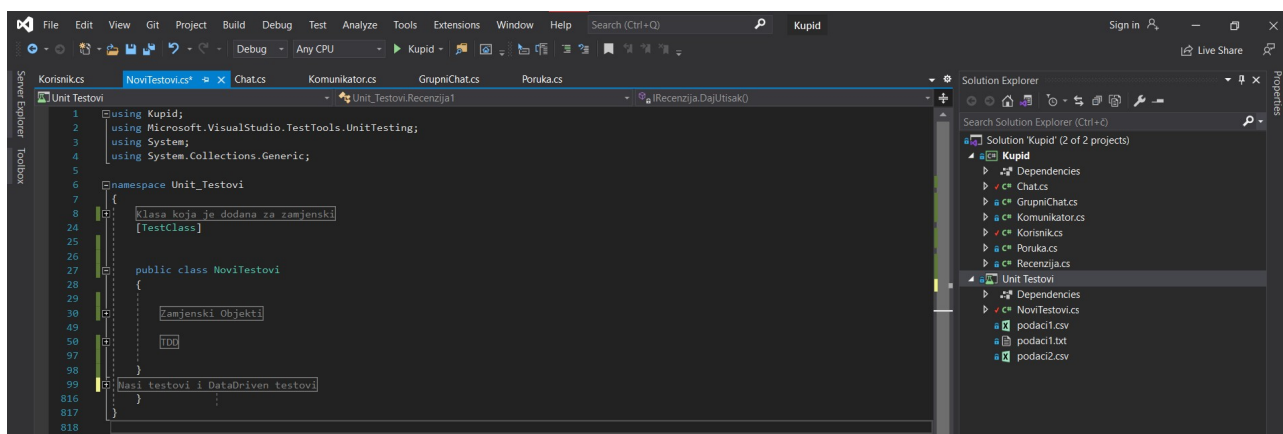
Ime i prezime: Amar Bešlagić

Broj indexa: 17047

GitHub link: https://github.com/MedinaKapo/Zadaca2_finalno

xml i csv file-ovi nisu se mogli uspješno kreirati, tako da su svi data-driven testovi napisani uz pomoć instrukcija iz poglavlja 3. laboratorijske vježbe 3 (DynamicData).

Također, svi data driven testovi su kreirani u okviru testnog projekta Unit Testovi i u okviru klase NoviTestovi.cs te u okviru testne klase NasiTestoviIDataDrivenTestovi (prebacivanje u drugi projekat je pravilo probleme i nije se mogla prepoznati ni jedna klasa iz rješenja i pored povezivanja sa referencom). Dakle, svi naši kreirani testovi, podaci za data-driven, kreirani zamjenski objekat se nalaze na istom mjestu gdje i svi testovi koji su došli u postavci zadaće, samo što su razdvojeni regionom sa drugim nazivom:



Razvijene funkcionalnosti:

Korisnik.PromjenaParametara – Medina Kapo

Poruka.IzracunajKompatibilnostKorisnika – Medina Kapo

Chat.DodajNovuPoruku – Sara Makešoska- Džebo

Komunikator.IzlistavanjeSvihPorukaSaSadrzajem – Amar Bešlagić

+razvijena je funkcionalnost, koja je bila potrebna za TDD, da bi prošli već definisani Unit testovi.
+Komunikator.SpajanjeKorisnika – Medina Kapo

```
}  
//Medina Kapo  
public void SpajanjeKorisnika()  
{  
    Korisnik k1 = new Korisnik();  
    if (korisnici.Count != 0)  
        k1 = korisnici[0];  
    else throw new ArgumentException("greska");  
    Korisnik k2 = new Korisnik();  
    for (int i = 1; i < korisnici.Count; i++)  
    {  
        k2 = korisnici[i];  
        if ((k1.Lokacija == k2.Lokacija && k1.ZeljenaLokacija == k2.ZeljenaLokacija && k1.Godine == k2.Godine) ||  
            (k1.Lokacija == k2.Lokacija && k1.ZeljenaLokacija == k2.ZeljenaLokacija) || (k1.Lokacija == k2.Lokacija &&  
            k1.Godine == k2.Godine))  
        {  
            Chat novi = new Chat(k1, k2);  
            razgovori.Add(novi);  
        }  
        else  
        {  
            throw new ArgumentException("greska");  
        }  
    }  
}
```

+kreiran je zamjenski objekat, da bi se mogli izvršiti testovi koji su vezani za dio programskog rješenja koji nije razvijen – Medina Kapo

```
{  
    #region Klasa koja je dodana za zamjenski  
    //Medina Kapo  
    /*kreirana je klasa Recenzija1.  
    * Ova klasa treba da sadrzi implementaciju metode DatjUtasak(), koja treba da vraća vrijednost "Pozitivan", ukoliko  
    *zelimo da nam prodje definisani test.*/  
  
    public class Recenzija1 : IRecenzija  
    {  
  
        string IRecenzija.DajUtasak()  
        {  
            return "Pozitivan";  
        }  
    }  
    #endregion  
    [TestClass]  
}
```

INSPEKCIJA FUNKCIONALNOSTI KOJE SU RAZVIJENE

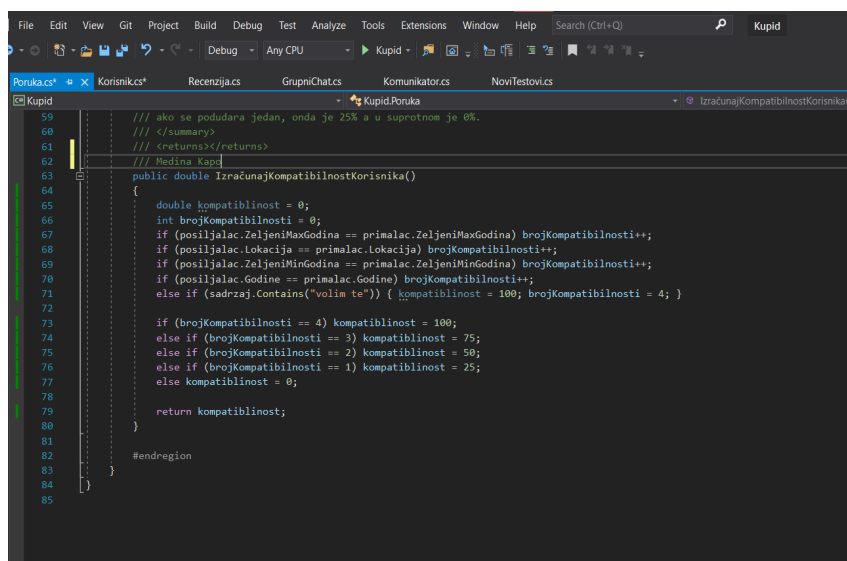
Korisnik.PromjenaParametara

```
/// </summary>  
/// <param name="slično"></param>  
///  
/// Medina Kapo  
public void PromjenaParametara(bool slično)  
{  
    if (slično)  
    {  
        zeljenaLokacija = lokacija;  
        zeljeniMinGodina = godine - 2;  
        ZeljeniMaxGodina = godine + 2;  
    }  
    else  
    {  
        while (true)  
        {  
            Random random = new Random();  
            Type type = typeof(Lokacija);  
  
            Array values = type.GetEnumValues();  
  
            int index = random.Next(values.Length);  
            Lokacija value = (Lokacija)values.GetValue(index);  
  
            if (value != lokacija)  
            {  
                zeljenaLokacija = value;  
                break;  
            }  
        }  
        zeljeniMinGodina = godine - 10;  
        ZeljeniMaxGodina = godine + 10;  
    }  
}
```

Ovaj kod nije naročito kompleksan i djeluje uredno. Ima pokriven slučaj ukoliko je slično true ili false, te na osnovu tog podatka izvršava definisane radnje. Prolaskom kroz petlju pronalazi se

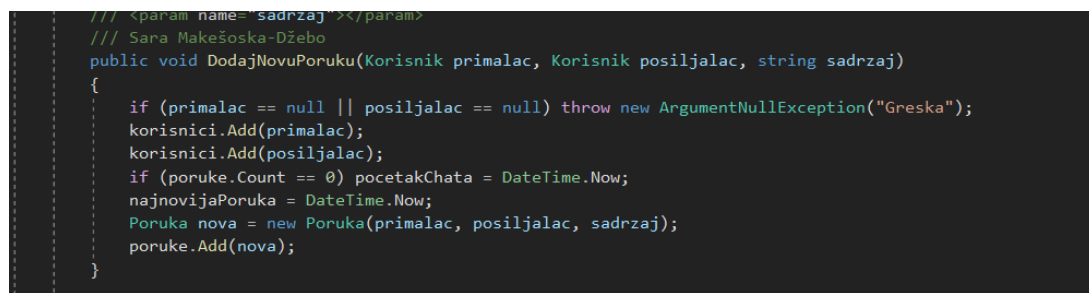
adekvatna lokacija, čime se ispunjava uslov koji nalaže da lokacija treba biti bilo koja druga osim stvarne lokacije.

IzracunajKompatibilnostKorisnika



Ovaj kod također nije naročito kompleksan, te se doima kako su svi slučajevi pokriveni, stoga za bilo koju vrijednost atributa će dati odgovor. U zavisnosti od toga kakvi su parametri kod učesnika u komunikaciji, moguće je doći do iznosa kompatibilnosti. Nije svakako moguće proslijediti null vrijednost za vrijednost koju imaju korisnici u okviru ove klase (konstruktor to onemogućava).

DodajNovuPoruku



Ovaj kod je poprilično jednostavan i u ovom kodu se vodi računa o tome ukoliko proslijeđeni korisnik bude null vrijednost.

Tipičan testni slučaj bi bio da se šalju validni podaci za korisnike i sadržaj poruke i to će da rezultira uspješnim kreiranjem poruke i dodavanjem iste u kolekciju poruka, te ukoliko se pošalje da su primalac ili pošiljalac jednaki null, tada se baca izuzetak.

IzlistavanjeSvihPorukaSaSadrzajem

```
Poruka.cs  Korisnik.cs  GrupniChat.cs  Komunikator.cs  Recenzija.cs  NoviTestovi.cs
Kupid.Komunikator
DaLiJeSpajanjeUspjesno(Chat c, IRecenzija r)

/// <summary>
/// Metoda u kojoj se vrši pronalazak svih poruka koje u sebi sadrže traženi sadržaj.
/// Ukoliko je sadržaj prazan ili ne postoji nijedan chat, baca se izuzetak.
/// U razmatranje se uzimaju i grupni, i individualni chatovi, a ne smije se uzeti u razmatranje
/// nijedan chat u kojem se nalazi korisnik sa imenom "admin".
/// </summary>
/// Asan Bešliagić
public List<Poruka> IzlistavanjeSvihPorukaSaSadrzajem(string sadržaj)
{
    if (sadržaj == null) throw new ArgumentNullException("greska");
    if (razgovori.Count == 0) throw new ArgumentNullException("greska");
    List<Poruka> vracam = new List<Poruka>();
    for (int i = 0; i < razgovori.Count; i++)
    {
        for (int t = 0; t < razgovori[i].Korisnici.Count; t++)
        {
            if (!razgovori[i].Korisnici[t].Equals("admin"))
            {
                List<Poruka> lokalne = razgovori[i].Poruke;
                for (int j = 0; j < lokalne.Count; j++)
                {
                    if (lokalne[j].Sadržaj.Contains(sadržaj) && !lokalne[j].Posiljalac.Ime.Equals("admin") &&
                        !lokalne[j].Primalac.Ime.Equals("admin") && !vracam.Contains(lokalne[j]))
                    {
                        vracam.Add(lokalne[j]);
                    }
                }
            }
        }
    }
    if (vracam.Count == 0) throw new ArgumentNullException("greska");
    return vracam;
}

public bool DaLiJeSpajanjeUspjesno(Chat c, IRecenzija r)
{
}
```

Ovaj kod ima pokriven i slučaj kada je sadržaj prazan, kao i kada uopšte nema razgovora. Kod se doima uredan i ukoliko se radi sa korisnicima koji nisu admin, pokušava se pronaći poruka koja u sebi sadrži odgovarajući sadržaj. Ukoliko takve poruke nema, baca se izuzetak. Funkcija “zaobilazi” korisnike koji su admin i ukoliko se desi da je jedan od učesnika u komunikaciji “admin”, takav razgovor se preskače.

Pokrivenost koda

Test Analyze Tools Extensions Window Help Search (Ctrl+Q) Kupid Sign in Live Share Buy me a coffee

Any CPU Kupid

Fine Code Coverage Coverage Summary Risk Hotspots Rate & Review Log Issue/Suggestion

Collapse all Expand all Filter:

Name	Covered	Uncovered	Coverable	Total	Line coverage
Kupid	252	3	255	590	98.8%
Chat	35	0	35	97	100%
GrupniChat	17	0	17	42	100%
Komunikator	89	2	91	159	97.8%
Korisnik	78	1	79	190	98.7%
Poruka	31	0	31	84	100%
Recenzija	2	0	2	18	100%
Unit Testovi	472	7	479	2451	98.5%