

Univerzitet u Sarajevu
Elektrotehnički fakultet Sarajevo
Odsjek za računarstvo i informatiku

Dokumentacija

ZADAĆA 2

Predmet: Verifikacija i validacija softvera

Studenti:

- Lino Bevanda (18048)
- Kerim Kadušić (18268)
- Matej Talirević (17749)

Grupa: 3

Sarajevo, novembar 2019.

Zadatak 1

- a) Metoda `povecajKapacitet` je testirana u sklopu testne klase `SekretaricaTest`. Testovima su obuhvaćeni svi mogući slučajevi povećanja sa i bez popusta na količinu. `StringAssert` je korišten za poređenje return vrijednosti te metode sa očekivanom vrijednosti. Izuzeci koji se mogu pojaviti uslijed poziva metode "naplati" vezani za stanje računa i broj kreditne kartice su testirani u sklopu dijela pod c). U konačnici je postignuta pokrivenost metode od 100%.

Sekretarica(Korisnik)	100%	0/9
povecajKapacitet(int)	100%	0/12

- b) Metode sa najvećim metrikama:

- Unutar klase `Sekretarica.cs` metoda sa najvećom Halstead metrikom je `povecajKapacite(int)` t za koju je i postignuta potpuna pokrivenost.
Izracunate metrike: $N1=27$, $N2=34$, $n1=8$, $n2=16$, $N=61$, $n=24$, $V=279.68$, $D=8.5$, $L=0.117647$, $E=2377.303$, $T=132.07$
- Unutar klase `Centrala.cs` metoda sa najvećom Halstead metrikom je `posaljiPoruku(Poruka p)` za koju je i postignuta potpuna pokrivenost.
Izracunate metrike za metodu `posaljiPoruku`: $N1=22$, $N2=24$, $n1=8$, $n2=11$, $V=195.404$, $D=8.72$, $L=0.1145833$, $E=1705.3498$, $T=94.74165$, $B=0.0475$

▼ Centrala	95%	3/62
▶ SekretariceNaMrezi	100%	0/6
Centrala(string)	100%	0/5
registrujSekretaricu(Sekretarica)	100%	0/10
posaljiPoruku(Poruka)	100%	0/35

- Za najlošiju vrijednost za metriku informacijske dobiti vrijedi da u klasi `Centrala` metoda `posaljiPoruku` ima najlošiju vrijednost
($Fanin=0(\text{poziva})+1(\text{čitanje})$), $Fanout=3(\text{pisanja})+3(\text{poziva}) \Rightarrow IFC=(6*1)^2=36$), kao što je navedeno ova metoda ima najveću pokrivenost
 - Za klasu `Sekretarica`, `arhivirajPoruke` ima najlošiju vrijednost ($Fanin=0(\text{poziva})+2(\text{čitanja})$), $Fanout=5(\text{poziva})+2(\text{pisanja}) \Rightarrow IFC=(7*2)^2=196$
- c) Svi izuzeci su testirani i nalaze se u klasama testova koje pripadaju klasama u čijim se metodama takvi izuzeci mogu pojaviti.

d) Prikaz podataka u CSV i XML fajlu:

1	ime,prezime,brojTelefona,datumRodjenja,brojKK,stanje
2	Matej,Talirevic,999999,25.02.1998,0,1000
3	Kerim,Kadusic,999888,12.06.1999,0,1000
4	Lino,Bevanda,777888,20.07.1998,0,1000

	1	2	3	4	5	6
	ime	prezime	brojTelefona	datumRodjenja	brojKK	stanje
1	Matej	Talirevic	999999	25.02.1998	0	1000
2	Kerim	Kadusic	999888	12.06.1999	0	1000
3	Lino	Bevanda	777888	20.07.1998	0	1000
4						

```
<?xml version="1.0" encoding="utf-8" ?>
<tests>
  <korisnik>
    <ime>Matej</ime>
    <prezime>Talirevic</prezime>
    <brojTelefona>999999</brojTelefona>
    <datumRodjenja>25.02.1998</datumRodjenja>
    <brojKK>0</brojKK>
    <stanje>1000</stanje>
  </korisnik>

  <korisnik>
    <ime>Kerim</ime>
    <prezime>Kadusic</prezime>
    <brojTelefona>999888</brojTelefona>
    <datumRodjenja>12.06.1999</datumRodjenja>
    <brojKK>0</brojKK>
    <stanje>1000</stanje>
  </korisnik>

  <korisnik>
    <ime>Lino</ime>
    <prezime>Bevanda</prezime>
    <brojTelefona>999888</brojTelefona>
    <datumRodjenja>12.06.1999</datumRodjenja>
    <brojKK>0</brojKK>
    <stanje>1000</stanje>
  </korisnik>
</tests>
```

Zadatak 2





















- a) Za metodu prioritetnog algoritma predviđena su dva testa koji će omogućiti uspješnu implementaciju prateći solid princip. Prvi test `PrioritetniAlgoritamTest1()` provjerava standardni slučaj tj. Ukoliko imamo poruke različitih prioriteta, a zatim drugi test provjerava ukoliko se desi da imamo poruke istih prioriteta. Obzirom da korisnik nije osigurao bacanje izuzetka ukoliko se proslijedi null object ili prazan niz dodan je if uslov koji to provjerava i baca izuzetak u tom slučaju. Postupak kreiranja metode (RFC TDD):

- Pozivam test kojim ću pozvati metodu za prioritetni algoritam (sa smislenim parametrima, poruke sa prioritetima)
- Test pada što je očekivano jer nisu implementirane funkcionalnosti
- Dodajem funkcionalnost za izbacivanje elementa sa najvećim prioritetom
- Pokrecem prethodni test, test prolazi
- Zatim posmatram rubni slučaj, šta ako pošaljem listu poruka sa istim prioritetima? Da li ću dovesti program u nekonzistentno stanje? Ispravnost provjeravam sa novim testom.
- Test prolazi.
- Sto znaci da trenutne funkcionalnosti zadovoljavaju zahtjeve.

Objasnjenje metode: kao element najveceg prioriteta se uzima prvi element u nizu, te se kroz petlju provjerava koji element ima veci prioritet. Nakon sto petlja prođe kroz niz trebali bi imati element najvećeg prioriteta kojeg treba poslati.

- b) Razvijen je zamjenski objekat tipa fake kojem je svrha da mijenja klasu `Poruka`, simulirajući prioritet poruke koji je jedini važan za testiranje prioritetnog algoritma. Nalazi se unutar testne klase `SekretaricaTest`. Klasa zamjenskog objekta je naslijeđena iz same klase `Poruka` da bi se mogla slati u relevantne metode, koje se testiraju, bez mijenjanja koda. Pozivom metode "prioritet" prioritet se uvećava za jedan i dodjeljuje u property, što olakšava testiranje raznih slučajeva prioriteta za algoritam.

Finalna pokrivenost:

▼  Total	89%	76/676
▼  TestProject	93%	22/295
▼  TestProject	93%	22/295
▶  CentralaTest	100%	0/42
▶  SekretaricaTest	93%	12/179
▶  KarticaTest	87%	7/53
▶  PorukaTest	86%	3/21
▼  Zadaca2	92%	30/357
▼  Zadaca2a.Izuzeci	100%	0/12
▶  AlgorithmNotValidException	100%	0/3
▶  CreditCardNotValidException	100%	0/3
▶  MessageTooLongException	100%	0/3
▶  NotEnoughMoneyException	100%	0/3
▼  Zadaca2a	97%	7/211
▶  Sekretarica	97%	4/149
▶  Centrala	95%	3/62
▼  Zadaca2a.Klase	83%	23/134
▶  Poruka	93%	3/42
▶  Kartica	83%	8/47
▶  Korisnik	73%	12/45