

Refaktoring koda

Decompose a boolean expression (Rastaviti logički izraz)

Prije:

```
Izbori.DodajKandidata(Ime.Text, Prezime.Text, MStanovanja.Text, jmb,  
(Nacionalnost.SelectedIndex==0 ? Kandidat.Nacionalnost.Bosnjak : Nacionalnost.SelectedIndex == 1 ? Kandidat.Nacionalnost.Hrvat : Nacionalnost.SelectedIndex==2 ? Kandidat.Nacionalnost.Srbin :Kandidat.Nacionalnost.Ostali));
```

Poslije:

```
Kandidat.Nacionalnost kn=new Kandidat.Nacionalnost();  
switch(Nacionalnost.SelectedIndex)  
{  
    case 0:  
        kn = Kandidat.Nacionalnost.Bosnjak;  
        break;  
    case 1:  
        kn = Kandidat.Nacionalnost.Hrvat;  
        break;  
    case 2:  
        kn = Kandidat.Nacionalnost.Srbin;  
        break;  
    case 3:  
        kn = Kandidat.Nacionalnost.Ostali;  
        break;  
}  
Izbori.DodajKandidata(Ime.Text, Prezime.Text, MStanovanja.Text, jmb, kn);
```

Rename a variable with a clearer or more informative name (Preimenovati varijablu sa više informativnim imenom)

Prije:

```
public static void DodajGlasaca(Glasac g, GlasackoMjesto gm)  
{  
    foreach (Glasac glasac in glasaci)  
        if (g.Jmbg == glasac.Jmbg) throw new Exception("Glasač je već unesen");  
    foreach (GlasackoMjesto glasmj in GMjesta)  
        if (glasmj.LokacijaMjesta == gm.LokacijaMjesta)  
        {  
            gm.DodajNovogGlasaca(g);  
            goto nastavak;  
        }  
    throw new Exception("Glasačko mjesto je ne postoji");  
    nastavak:  
    Glasaci.Add(g);  
}
```

Poslije:

```
public static void DodajGlasaca(Glasac noviGlasac, GlasackoMjesto glasackoMjestoGlasaca)  
{  
    foreach (Glasac glasac in glasaci)  
        if (noviGlasac.Jmbg == glasac.Jmbg) throw new Exception("Glasač je već unesen");  
    foreach (GlasackoMjesto glasackoMjesto in GMjesta)  
        if (glasackoMjesto.LokacijaMjesta == glasackoMjestoGlasaca.LokacijaMjesta)  
        {  
            glasackoMjestoGlasaca.DodajNovogGlasaca(noviGlasac);  
            goto nastavak;  
        }  
    throw new Exception("Glasačko mjesto je ne postoji");  
    nastavak:  
    Glasaci.Add(noviGlasac);  
}
```

Replace a magic number with a named constant (Zamijeniti 'magične' brojeve sa imenovanom konstantom)

Prije:

```
public long Jmbg { get => jmbg; set => jmbg = (value < 1000000000000 || value > 999999999999) ? throw new Exception("Pogrešan JMBG") : value; }
```

Poslije:

```
private const long gornjaGranicaJMBG = 1000000000000;  
private const long donjaGranicaJMBG = 999999999999;
```

```
public long Jmbg { get => jmbg; set => jmbg = (value < gornjaGranicaJMBG || value > donjaGranicaJMBG) ? throw new Exception("Pogrešan JMBG") : value; }
```

Design patterni

Kreacijski pattern

U ovom projektu bi dobro bilo koristiti Singleton kreacijski pattern za predstavljanje klase izbora jer u jednom sistemu nema potrebe za više klasa izbora zbog toga što je aplikacija predviđena za samo jedne izbore.

Strukturni pattern

U ovom projektu dobro bi bilo koristiti strukturni pattern Facade za predstavljanje klase izbora, jer se ovaj pattern koristi da sakrije podsistem. Interakcija se vrši samo sa klasom izbora i ta klasa obavlja sve operacije.

Pattern ponašanja

U ovom projektu bi bilo dobro koristiti Iterator pattern ponašanja za prolazak kroz sve kandidate ili političke subjekte unutar klase izbora.