

O Visual studiu

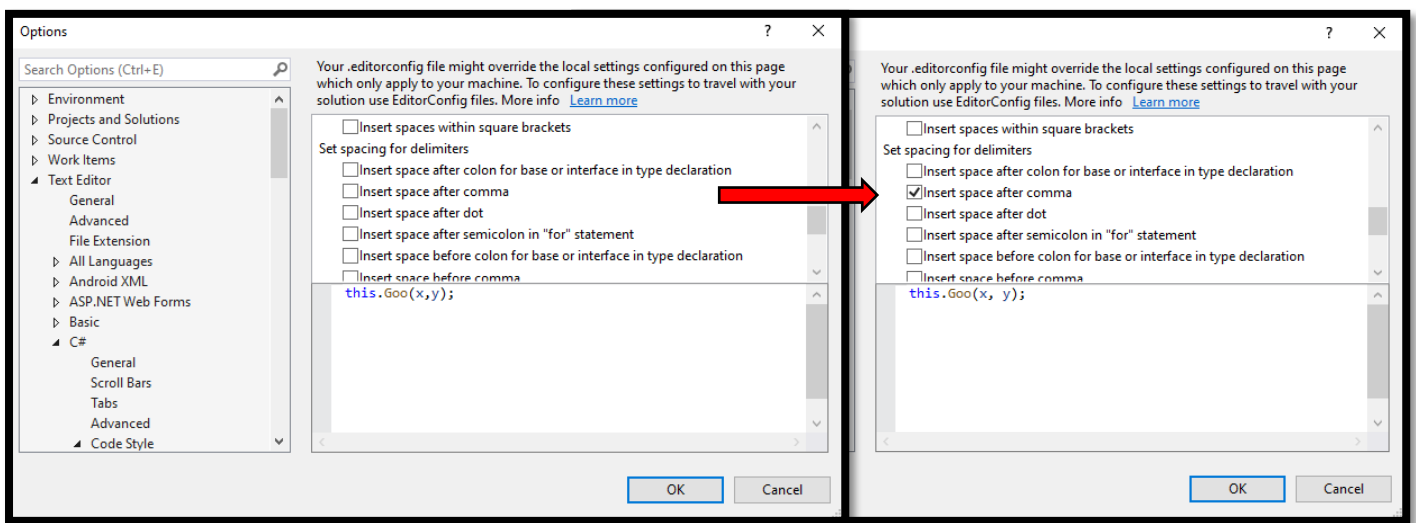
V teh navodilih bom predstavila, kako lahko v Visual studiu 2019 nastavimo obnašanje urejevalnika kode glede na svoje želje. Osredotočila sem se na avtomatsko pisanje presledkov med pogoji in operatorji, česar se tudi pri »ročnem« pisanju kode vedno držim. Zapis na tak način omogoči boljši pregled in jasnejše branje.

Vse nastavitve za poljubno obnašanje urejevalnika najdemo v **Tools/Options/TextEditor**. Te, ki pa so potrebne za nastavitve z presledki, pa najdemo v **Tools/Options/TextEditor/C#/Code Style/Formatting/Spacing**.

Presledki za vejicami, podpičji ter med operatorji:

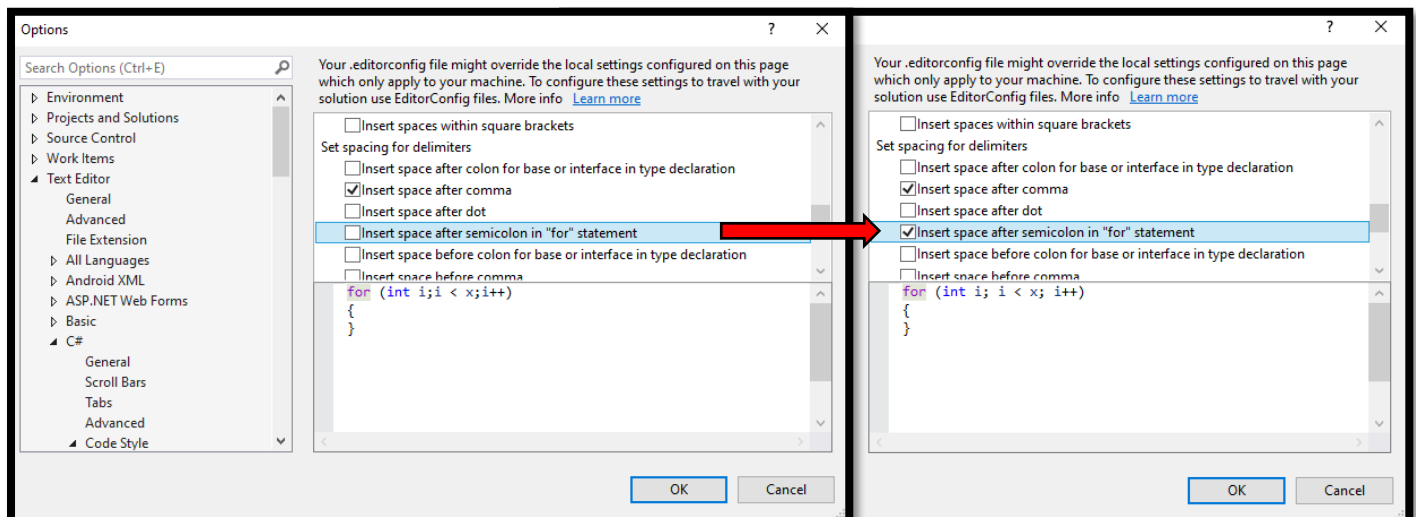
Da so pogoji (na primer pri if stavkih ali zanki for) ter parametri (pri metodah in funkcijah) bolj jasni in opazni, je dobro, da jih ločimo s presledki. Da pa se ročnemu pisanju izognemo, lahko v nastavitvah Visual Studia to nastavimo na avtomatsko urejanje.

Če želimo, da za vsako vejico sledi presledek, je potrebno obkljukati možnost *Insert space after comma*.



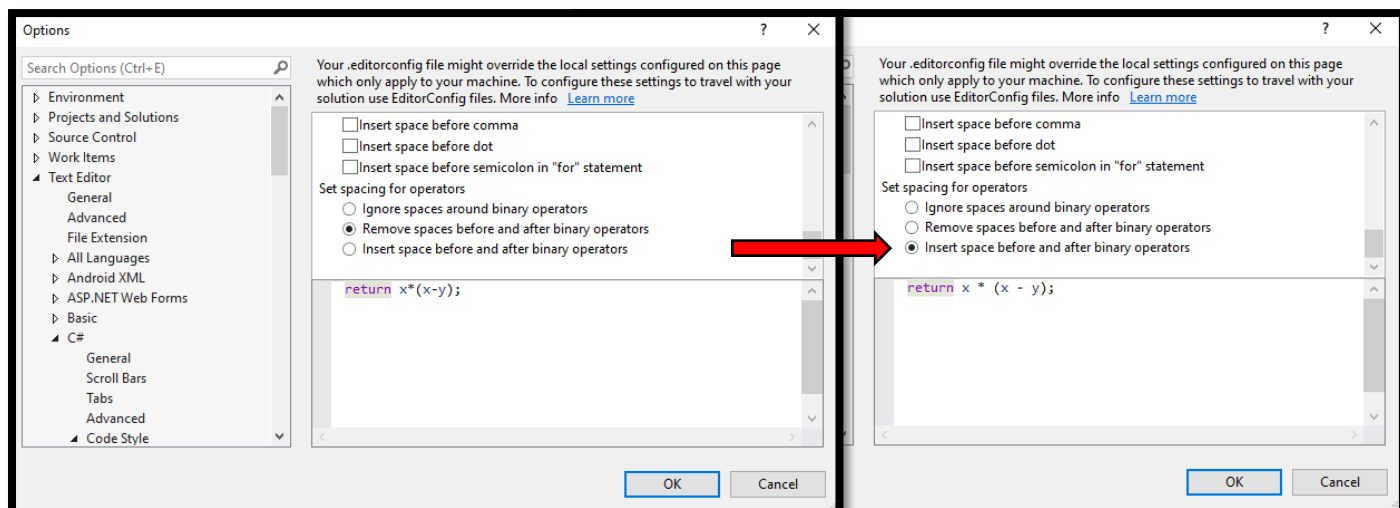
Slika 1: zapis brez presledkov za vejicami se spremeni v zapis s presledki.

Če želimo, da med pogoji v for zanki (pomeni za podpičji) sledi presledek, je potrebno obkljukati *Insert space after semicolon in "for" statement*.



Slika 2: zapis brez presledkov za podpičji se spremeni v zapis s presledki za podpičji.

Če želimo, da se med operatorji samodejno ustvarijo presledki, je potrebno izbrati *Insert space before and after binary operators*.



Slika 3: zapis brez presledkov med operatorji se spremeni v zapis s presledki med operatorji.

Spodaj levo je primer metode, ki ni opremljena s presledki. Desno pa je potem ista metoda, ki je urejena po zgoraj omenjenih nastavitvah.

```
static int TestnaFunkcija(int n)
{
    int vrednost=0;
    for(int i=0;i<n;i++)
    {
        if (i<3&& i!=0)
        {
            vrednost=vrednost+i*7-2;
        }
        else
        {
            vrednost=vrednost-i*2;
        }
    }
    return vrednost*2+1;
}
```

```
static int TestnaFunkcija(int n)
{
    int vrednost = 0;
    for(int i = 0; i < n; i++)
    {
        if (i < 3 && i != 0)
        {
            vrednost = vrednost + i * 7 - 2;
        }
        else
        {
            vrednost = ..vrednost - i * 2;
        }
    }
    return vrednost * 2 + 1;
}
```

Opazimo lahko, da je z uporabo presledkov koda nekoliko bolj pregledna in lepša za branje.