

Univerza v Ljubljani
Fakulteta za *matematiko in fiziko*



Snippets v Visual Studio

PROGRAMIRANJE 3

Avtor naloge:

Gal Zakrajšek

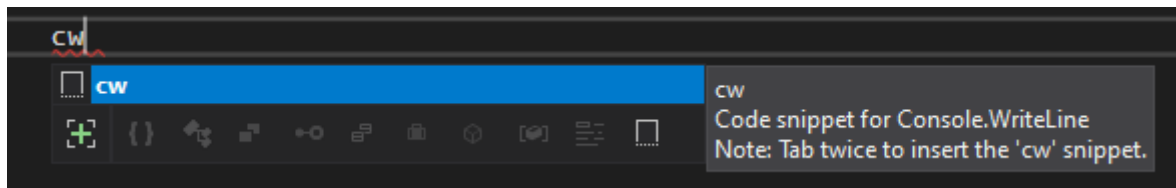
Ljubljana, 2022

Kazalo

1. Ideja in uporabnost snippetov v Visual Studiu.....	3
2. Vrste snippetov	3
2.1. Preproste snippete.....	3
2.2. Obdajajoče snippete	3
2.3. Snippete z parametri.....	3
3. Ustvarjanje svojih snippetov	4
3.1. Kreiranje XML datoteke	4
3.2. Sintaksa preprostih snippetov	5
3.3. Uvoz snippetov v Visual Studio	6
4. Kreiranje snippeta z parametri	7
5. Dodatne nastavitve	8
5.1. Postavitev kurzorja po generiranju kode	8
5.2. Importi v snippetih.....	8
6. Viri	9

1. Ideja in uporabnost snippetov v Visual Studio

Snippeti, so delčki kode, ki si jih uporabnik lahko shrani pod določeno kratico in se tako izogne večkratnemu pisanju enake kode. Kot primer, si lahko pogledamo snippet "cw". Uporabnik velikokrat želi napisati nekaj v konzolo in pri jeziku C#, se to naredi z vrstico "Console.WriteLine()". Da se uporabniku olajša delo, je v Visual Studio že vnaprej določen snippet "cw", ki omogoča, da takoj, ko v vrstico napiše črki "cw" in za tem dvakrat pritisne tipko "tab", Visual Studio sam izpolni vrstico z celotnim ukazom "Console.WriteLine()" in mu tako prihrani čas pri pisanju. Seveda snippeti niso omejeni samo na eno vrstico, v njih lahko shranimo celotne zanke, bloke, metode, razrede. Takoj, ko uporabnik napiše kratico v Visual Studio ta že sam pove v pojavnem oknu, da za to kratico stoji snippet. Pojavno okno izgleda sledeče:



Snippeti pa ne izboljšajo samo hitrosti pisanja. Z uporabo se izognemo tudi tipkarskim napakam, saj bo snippet zagotovo vedno pravilno generiral kodo. Tako uporabnik ne porabi dodatnega časa za odkrivanje raznih hroščev, ker je pozabil na kakšno vrstico.

Tudi v primerih, kjer na istem projektu dela več koderjev snippeti pridejo prav, saj lahko z njimi postavimo neke standarde o izgledu kode, tako bo na vseh mestih za enako opravilo uporabljena podobna koda in iste knjižnice.

2. Vrste snippetov

Snippete delimo na več vrst glede na to, kakšen tip kode nam generirajo. Delimo jih na:

2.1. Preproste snippete

Preprosti snippeti so takšni, ki nam samo vstavijo vrstico ali več, ki je potem uporabnik ne spreminja. Tak snippet je na primer "cw", ki smo ga omenili zgoraj.

2.2. Obdajajoče snippete

Obdajajoči snippeti nam generirajo vrstice, kjer nam pustijo prazne bloke { }, kamor potem uporabnik sam vnese željeno kodo. Primer takega snippeta je "try", ki nam generira try, catch stavek in pri vsakem prazen blok, katere izpolnimo sami. Drug primer je tudi "if", ki generira if, else stavek.

2.3. Snippete z parametri

Snippet z parametri pa so posebni, ko generirajo kodo kurzor postavijo na vnaprej določen parameter. Tako ga lahko takoj spremenimo in Visual Studio sam poskrbi, da se v celotni generirani

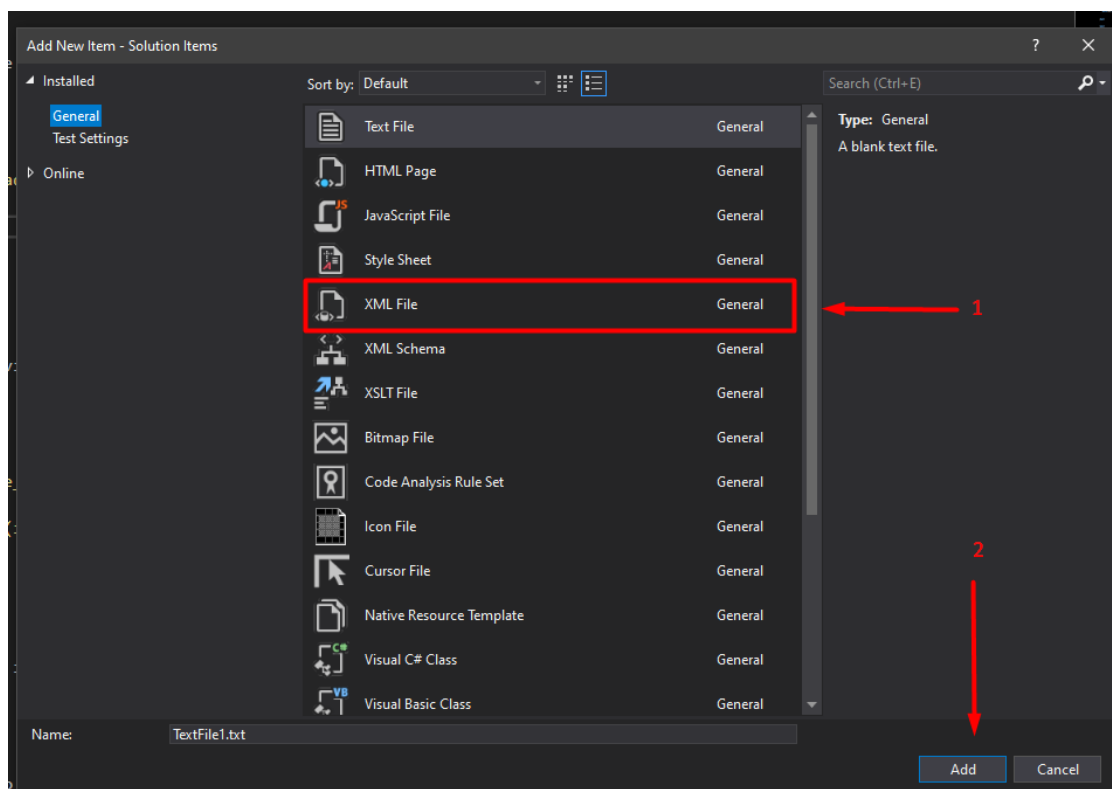
kodi to ime spremenljivke preimenuje. Primer takega snippeta je "for", ki nam ustvari for zanko in kurzor takoj postavi na mesto, kjer je deklariran števec, ki je privzeto označen z črko i. Ko bomo i spremenili v poljubno ime, se bo ta spremenil tudi pri drugih delih zanke. Na sliki je vidno, kaj vse se bo spremenilo, ko preimenujemo števec i.

```
for (int i = 0; i < length; i++)  
{  
    ...  
}
```

3. Ustvarjanje svojih snippetov

3.1. Kreiranje XML datoteke

Zdaj smo si pogledali, kaj so snippeti in zakaj so uporabni. Zdaj pa si pogledjmo, kako se ustvari lastne snippete in uvozi v Visual Studio. Snippet se piše v strukturi imenovani xml, zato bomo za začetek ustvarili novo xml datoteko. V Studiu ustvarimo nov projekt in ko se nam generirajo vse datoteke bomo z desnim klikom na "solution" izbrali Add -> New item.... Odpre se nam okno, za ustvarjanje novih datotek v naši rešitvi. Tam izberemo "XML file" in potrdimo izbiro. Okno, kjer izbiramo izgleda tako:



Ustvari se nam prazna xml datoteka, kjer bomo shranjevali naše snippete.

3.2. Sintaksa preprostih snippetov

V prazno datoteko, bomo prilepili spodnjo kodo, ki je osnovna sintaksa za snippete:

```
<?xml version="1.0" encoding="utf-8"?>
<CodeSnippets xmlns="http://schemas.microsoft.com/VisualStudio/2005/CodeSnippet">
  <CodeSnippet Format="1.0.0">
    <Header>
      <Title></Title>
      <Author></Author>
      <Description></Description>
      <Shortcut></Shortcut>
    </Header>
    <Snippet>
      <Code Language="CSharp">
        <![CDATA[]]>
      </Code>
    </Snippet>
  </CodeSnippet>
</CodeSnippets>
```

Zdaj je naša naloga, da izpolnimo zgornjo kodo. Vsak snippet ima podatke o avtorju in pa opis, ki jih bo Visual Studio prikazal ob uporabi. Zato začnemo z izpolnjevanjem teh podatkov med značke xml-ja.

1. **Title:** Naslov snippeta, ki bo prikazan v seznamu snippetov
2. **Author:** Ime in priimek avtorja tega snippeta
3. **Description:** Kratak opis tistega, kar bo generiral snippet
4. **Shortcut:** Kratica, ki jo mora uporabnik napisati, da se generira naš snippet

Tako smo izpolnili vse podatke o našem snippetu. Vse kar nam manjka so še vrstice kode, ki jih želimo generirati. Te bomo napisali med oglate oklepaje pri vrstici <![CDATA[tukaj]]>. Snippet bo generiral točno takšno kodo, kot jo bomo napisali tukaj. Upošteval bo vse presledke, prehode v novo vrstico in tabulatorje.

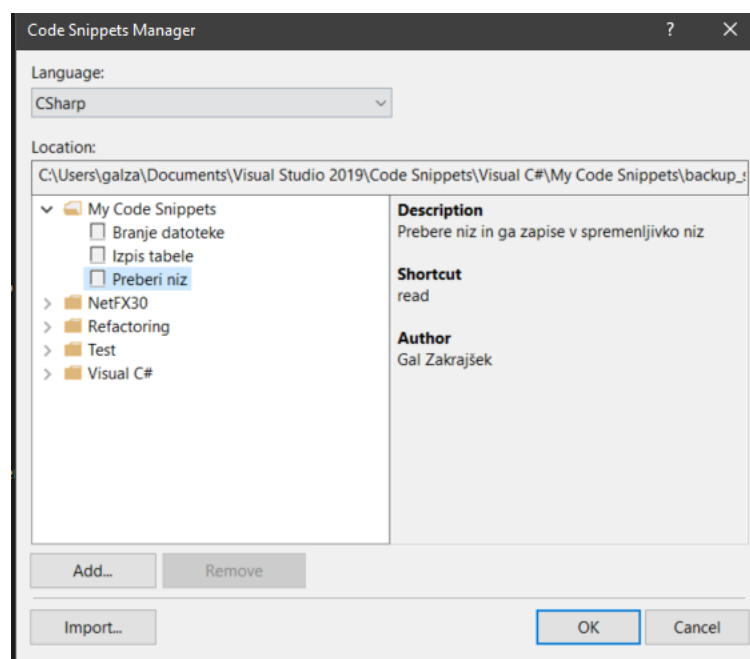
Za primer si pogledajmo kako izgleda, če želimo ustvariti snippet "cr", ki nam bo generiral vrstico v C#, ki čaka, da uporabnik v konzolo nekaj napiše. Končni snippet bo izgledal tako:

```
<?xml version="1.0" encoding="utf-8"?>
<CodeSnippets xmlns="http://schemas.microsoft.com/VisualStudio/2005/CodeSnippet">
  <CodeSnippet Format="1.0.0">
    <Header>
      <Title>Preberi niz</Title>
      <Author>Gal Zakrajšek</Author>
      <Description>Prebere niz in ga zapiše v spremenljivko niz</Description>
      <Shortcut>cr</Shortcut>
    </Header>
    <Snippet>
      <Code Language="CSharp">
        <![CDATA[string niz = Console.ReadLine();]]>
      </Code>
    </Snippet>
  </CodeSnippet>
</CodeSnippets>
```

Ko imamo celoten obrazec za snippet izpolnjen ga shranimo.

3.3. Uvoz snippetov v Visual Studio

Zdaj imamo našo xml datoteko pripravljeno za uvoz v Visual Studio. Pred uvozom je pomembno, da ročno spremenimo kratico naše xml datoteke iz .xml v .snippet. Ko smo preimenovali datoteko, odpremo ponovno Visual Studio in se pomaknemo v meni Tools -> Code Snippet Manager.... Notri zagledamo gumb Import, ki nam odpre okno v katerem poiščemo in izberemo našo .snippet datoteko. Ko končamo z uvozom, jih v Code Snippet Managerju vidimo pod rubriko "My Code Snippets". Znotraj bodo naštetni vsi snippeti, ki smo jih uvozili.



4. Kreiranje snippeta z parametri

Na začetku smo govorili tudi o snippetih z parametri, ki nam takoj ko kodo generira kurzor postavi na ime spremenljivke, da jo lahko na hitro preimenujemo in nato se bo v vseh vrsticah, ki jih je snippet generiral ta spremenljivka preimenovala. To dosežemo z nekaj dodatnimi vrsticami v naši xml datoteki. Dodatne vrstice, ki jih potrebujemo so naslednje

```
<Snippet>
  <Code Language="CSharp">
    <![CDATA[]]>
  </Code>
  <Declarations>
    <Literal>
      <ID></ID>
      <ToolTip></ToolTip>
      <Default></Default>
    </Literal>
  </Declarations>
</Snippet>
```

V naši kodi, ki jo želimo generirati označimo parameter tako, da ga obdamo z "\$" znaki. Tako bo Visual Studio vedel, da gre za parameter. Nato dodamo zgornje vrstice in jih izpolnimo sledeče:

1. **ID:** ime našega parametra. Tukaj ne navedemo "\$" znakov spredaj in zadaj.
2. **ToolTip:** Opis parametra, da uporabnik takoj ugotovi kaj predstavlja
3. **Default:** Privzeta vrednost, ki bo obstala, če je uporabnik ne bo prepisal.

To so vse dodatne vrstice, ki jih potrebujemo za nastavljanje parametrov. Za lažjo predstavbo sledi še primer snippeta, ki generira kodo za izpis tabele celih števil v eno vrsto. Kot parameter je tukaj ime tabele, ki jo bo uporabnik vnesel ob generiranju.

```
<Snippet>
  <Code Language="CSharp">
    <![CDATA[for (int i = 0; i < $ime_tabele$.Length; i++)
    {
        Console.WriteLine($ime_tabele${i} + (i + 1 < $ime_tabele$.Length? ", " : "\n"));
    }]]>
  </Code>
  <Declarations>
    <Literal>
      <ID>ime_tabele</ID>
      <ToolTip>Podaj ime tabele za izpis</ToolTip>
      <Default>tab</Default>
    </Literal>
  </Declarations>
</Snippet>
```

5. Dodatne nastavitve

Spoznali smo skoraj že vse nastavitve snippetov. Sledita samo še dve preprosti, ki pa lahko uporabniku še dodatno olajšata pisanje kode.

5.1. Postavitev kurzorja po generiranju kode

Snippet ima tudi možnost, da takoj, ko generira kodo prestavi kurzor na vnaprej določeno mesto. To dosežemo z rezervirano besedo "\$selected\$". Tako bo Visual Studio vedel, kje želimo naš kurzor, po generiranju kode. Primer uporabe je pri generiranju if stavka. Takrat ko ustvarimo if stavek želimo, da kurzor skoči med oklepaje za pogoj, saj bomo najverjetneje tam začeli pisati kodo.

5.2. Importi v snippetih

Včasih bomo v določenih snippetih uporabili kakšno knjižnico, ki ni privzeto dodana v naš program. Tudi za to imamo rešitev in sicer pri sintaksi, dodamo naslednje vrstice in tako bo snippet sam dodal določene knjižnice v naš program. V naslednjem primeru bi program dodal knjižnico "System.IO".

```
<Snippet>
  <Code Language="CSharp">
    <![CDATA[]]>
  </Code>
  <Imports>
    <Import>
      <Namespace>System.IO</Namespace>
    </Import>
  </Imports>
</Snippet>
```


6. Viri

- Code snippets. (30. April 2022). <https://docs.microsoft.com/en-us/visualstudio/ide/code-snippets?view=vs-2022> (Dostopno 23.8.2022)
- Walkthrough: Create a code snippet. (8. Junij 2022). <https://docs.microsoft.com/en-us/visualstudio/ide/walkthrough-creating-a-code-snippet?view=vs-2022> (Dostopno 23.8.2022)
- Snippets. <https://riptutorial.com/visual-studio/example/7913/snippets> (Dostopno 23.8.2022)