

## Naloga 4

Jaka Čop

24. marec 2025

**Za vsako od osmih Shneidermanovih zlatih pravil navedite lasten primer vmesnika, ki se drži posameznega pravila. Obrazložite! (Isti vmesnik lahko podate kot primer pri več pravilih.)**

Schneidermanovim pravilom sledi npr. večina modernih urejevalcev besedila. Moj najljubši in najbolj poznan urejevalec je **Visual Studio Code (VS Code)**, ki po mojem mnenju zadostuje **vsem** osmim pravilom. Pri vsakem pravilu bom navedel svojo utemeljitev za to izbiro. Obenem pa bom poskusil navesti tudi primer vmesnika, ki se določenega pravila ne drži.

1. Vse „običajne“ bližnjice delujejo tudi znotraj VS Code-a. Npr. Ctrl + C, Ctrl + V, Ctrl + X, Ctrl + Z, Ctrl + Y, ki kopirajo, lepijo, izrežejo, razveljavijo in ponovno naredijo; kot v ostalih programih – kar ni presenetljivo, saj je VS Code, tako kot Windows, razvil Microsoft in je verjetno zato z namenom konsistenten bližnjicam Windows-a. Ne spomnim se programa, v katerem navedene bližnjice ne bi delovale. Morda na Mac-ih, s katerimi pa nimam izkušenj.
2. Vseh bližnjic, ki jih VS Code nudi, ne bom verjetno nikoli poznal – verjetno jih ne bo nihče, saj jih je ogromno. Med zanimivimi naprednimi bližnjicami pa so: Ctrl + Alt + Shift + ArrowUp (ustvari kazalcev v več vrsticah, da lahko urejamo v več vrsticah hkrati – uporabno za brisanje enakih elementov), Ctrl + Shift + F5 (ponovno zažene razhroščevanje), Ctrl + Shift + P (prikaže vse možne ukaze in tudi bližnjice za nekatere – za ostale pa si lahko bližnjice celo poljubno nastavimo – možnosti je res skoraj neskončno). Za primerjavo sem v Beležnici (Microsoftovem programu) naštel vsega skupaj 19 bližnjic. Težko bi rekli, da ta program vsebuje bližnjice za napredne uporabnike – 19 se jih mnamreč lahko nauči vsakdo.
3. VS Code je v javljanju napak zelo dober. Na dnu programa se ob uspešni izvedbi programa pojavi kljukica. Ob napakah pa nam vrne opis napake in tudi vrstico, v kateri naj bi se napaka zgodila. Medtem pa nam Notepad (Beležnica) ne da skoraj nobene povratne informacije. Edina, ki jo dobimo, je ta, da v imenu datoteke izgine zvezdica, ko to datoteko shranimo. Bolj minimalističnega „feedback-a“ skoraj ne bi mogli dobiti.
4. Delno že odgovorjeno v prejšnji točki zato ponavljam, da je v VS Code zelo jasno, kdaj se program izvede uspešno in kdaj ne. Bodisi v zavihku „PROBLEMS“, bodisi v zavihku „TERMINAL“ lahko preberemo informacijo, da pri zagonu programa ni bilo težav. Prav tako lahko opazimo kljukico na dnu programa – tudi ob navadnem

shranjevanju datoteke in ne nujno njenemu zagonu. V zavihku „TERMINAL“ pa bomo ob uspešno zaključeni izvedbi našega programa našli tudi iskani/želeni rezultat.

5. Tudi na področju zmanjševanja napak je VS Code odličen. Besedilo/koda je vedno smiselno pobarvana po ukazih. Kar je še posebej koristno pri gnezdenju velikega števila oklepajev. Primer programa, ki tega sploh ne počne, pa je ponovno Beležnica, ki teksta ne sploh ne pobarva.
6. Enostaven preklic dejanj nudijo praktično vsi programi, ki podpirajo bližnjico Ctrl + Z (tudi Beležnica). Prav tako lahko v VS Code zlahka zaustavimo program, ki se na primer vrti v neskončnem krogu, npr. s klikom na „Kill Terminal“ (privzeta bližnjica: Ctrl + F4). Jezik  $\text{\LaTeX}$  pa je celo narejen tako, da ves čas shranjuje spremembe glavne datoteke v posebno datoteko, iz katere lahko kadarkoli preidemo nazaj na prejšnjo različico glavne datoteke (in je ne izbriše nikoli; Beležnica je sploh ne ustvari – nima možnosti za vrnitev po zaprtju programa; prav tako Word prej ali slej iz pomnilnika odstrani shranjene spremembe v glavni datoteki). Morda je  $\text{\LaTeX}$  glede enostavnega in predvsem možnosti poljubnega vračanja v starejše verzije datotek najboljši primer za 6. Schneidermanovo pravilo, a mu po mojem zadošča tudi VS Code.
7. Med pisanjem samega besedila/kode nam program VS Code vseskozi nudi smiselne predloge za nadalajevanje napisanega ukaza ali besede, a ne popravlja besedila sam od sebe, kot to počne „autocorrect“, temveč pusti, da uporabnik izbere katero ali nobene od ponujenih opcij. Notepad tega ne počne.
8. VS Code nam ponudi dopolnjevanje ukazov skupaj s kratkimi razlagami, kaj določeni ukazi naredijo. Obenem nam poda tudi informacijo, katere argumente ti ukazi sprejmejo. Npr. v  $\text{\LaTeX}$ -u ukaz `\begin{ime_okolja}` ustvari novo okolje. V zavite oklepaje pa spada ime okolja. Obe informaciji, skupaj s kratko razlago, nam takoj ponudi VS Code in se jih tako ne rabimo učiti na pamet. Beležnica tudi te funkcije (ali kakšne podobne) nima vgrajene.