

UNIVERZA V MARIBORU  
FAKULTETA ZA ELEKTROTEHNIKO,  
RAČUNALNIŠTVO IN INFORMATIKO

Gregor Perčič, Matic Lukežič

# **DRUGA SEMINARSKA NALOGA**

seminarska naloga

Maribor, januar 2022

# Kazalo

1	Uvod	1
2	Aritmetika	1
3	Pretvarjanje med številskimi sistemi	1
4	Logična vrata	2
5	Branje iz datoteke	3

## 1 Uvod

Kalkulator sva razvijala v programskem jeziku Python. Za ta jezik sva se odločila zato, ker je preprost in ima veliko različnih funkcij in knjižnic, ki so nama bile v pomoč pri izdelavi kalkulatorja.

Za izgled kalkulatorja oz. za grafični vmesnik sva uporabila knjižnico Tkinter. Vključena je v standardne namestitve Pythona. Z vključitvijo Tkinter knjižnice sva lahko v najino okno (okvir), dodajala gumbe in zaslone za izpis.

## 2 Aritmetika

Najprej je bilo treba napisati kodo za kalkulator, ki vsebuje osnovne računske operacije, kot so seštevanje, odštevanje, množenje in deljenje. Na srečo v jeziku Python obstaja funkcija `eval` (ang. evaluate), ki nam izračuna podan račun. Vhod v funkcijo je niz števil, funkcija pa nato izračuna in nam vrne vrednost niza. Torej v funkcijo ne dajemo navadnih števil, ampak jih moramo pretvoriti v tip nizov (ang. string). Ima tudi možnost prepoznavanja semantičnih napak, kot je na primer deljenje z nič.[1] Nato sva še dodala funkcije za računanje korenov, potenciranje ter modul ali deljenje s celoštevilskim ostankom. Python sam po sebi ne zna računati korenov ali potencirati števil, zato sva morala uvoziti knjižnico `math`. Na začetku sva imela samo eno vrstico za vnos računa ter izpis rezultata, in da bi kalkulator bil preglednejši, sva dodala še eno vrstico. Torej, ko uporabnik vnaša številke, jih program vnaša v spodnjo vrstico, kadar pa želi dobiti rezultat, klikne na gumb `=`, s katerim program izračuna vrednost računa oz. izraza, ga nato izpiše v zgornji vrstici, v spodnji pa se izpiše rezultat. Spremenila sva tudi, kako se račun prikazuje na zaslon. Primarno se vsaka posamezna črka vpisuje od leve proti desni, poravnava je v levo, midva pa sva naredila, da se številke vpisujejo od desne proti levi,[2] poravnava pa je v desno.

## 3 Pretvarjanje med številskimi sistemi

Slednjega sva se lotila takole: uporabnik vnese število, ki ga želi pretvoriti, in pritisne gumb `PRET`. Nato vnese bazo, iz katere želi pretvoriti število (kalkulator tudi preveri, ali je morda vnos glede na bazo neustrezen, in to javi v okence). Ponovno pritisne `PRET` ter vnese bazo,

v katero želi pretvoriti število (najin kalkulator ima implementirane baze od 2 do 16). Nato še zadnjič pritisne **PRET** in v okencu se mu izpiše rezultat.

Pri tej točki si je vredno pogledati nekaj konkretnih tehničnih ovir, na katere sva naletela. Za primer, morala sva emulirati zanko `do {...} while`, ki ni vgrajena v jeziku Python.[3] V glavnem je funkcija za pretvarjanje delovala na nizih, zato sva se morala pozanimati tudi o določenih metodah na nizih, ki jih ponuja Python.[4, 5] Najzahtevnejši del implementacije je bilo verjetno usklajevanje delovanja programa z vmesnikom. Ker se funkcija, lastna gumbu **PRET**, kliče večkrat, sva morala nekako shraniti stanje, ki je rezultat vsakega klica, ter ga naslednji klic na nek način uporabi. To sva naredila z rezervirano besedo `global`,[6] ki spremenljivko v funkciji spremeni v globalno spremenljivko, do katere lahko dostopajo tudi drugi klici funkcije. Natančneje, ko kliknemo gumb **PRET**, se sproži funkcija **pretvorba** prvič, in izpiše poizvedbo za bazo, iz katere pretvarjamo, poleg tega pa se shrani število, ki je bilo prej v okencu. Pri drugem kliku se shrani prva baza, v okencu pa se izpiše poizvedba za drugo bazo. Pri tretjem kliku se sproži funkcija **pretvori**, ki iz začetnega števila in obeh baz izračuna rezultat, ter ga izpiše v okencu.

Seveda je bilo potrebno rezultate testirati ter tako preveriti, ali je koda napisana pravilno. To sva storila z enim od spletnih pretvornikov med bazami.[7]

## 4 Logična vrata

Implementirala sva logične operacije **NEG**, **AND**, **OR**, **XOR**, **NOR**, **XNOR** in **NAND**, ki delujejo nad bazami od 2 do 16. Uporabnik brez predhodnega vnosa klikne na zeleno logično operacijo in izpiše se mu poizvedba za bazo, v kateri naj bodo argumenti logične operacije. Vnese zeleno bazo in ponovno klikne isti gumb. Vnese prvo število (če je operacija dvomestna) in ponovno klikne gumb. Potem vnese drugo število in klikne gumb **=**, ki ovrednoti logično operacijo glede na podana argumenta ter izpiše rezultat. Med operacijami je **NEG** posebnost, saj je ne ovrednotimo z **=**, pač pa trikrat kliknemo na njen gumb: prvič za vnos baze, drugič za vnos števila, in tretjič za ovrednotenje negacije.

Sprogramirala sva tudi samodejno dodeljevanje ničel na začetku krajšega od dveh števil (pri dvomestnih logičnih operacijah), da je možno ovrednotiti logični izraz s poljubno velikima številoma v argumentih. (Ker celoten postopek v ozadju deluje na podlagi bitnih polj, bi bilo seveda nemogoče ovrednotiti dve števili, ki v binarnem zapisu ne bi bili iste dolžine.) Programerska rešitev za logična vrata deluje precej podobno, kot rešitev pretvorbe med bazami. Znova sva se poslužila pogoste uporabe ključne besede `global`, novost pa je bila to, da se je izraz ovrednotil preko dveh gumbov namesto enega samega. Zato ima vsak od gumbov za logične operatorje nase vezano funkcijo `nalozi(k)`, ki nastavi globalno spremenljivko glede na `k`. Na gumb **=** je vezana funkcija `sprozi`, ki zažene primerno funkcijo za ovrednotenje in izpis logične operacije glede na nastavljeno globalno spremenljivko. Naj omeniva tudi, da je bila funkcija `pretvori` iz prejšnjega razdelka pri logičnih vratih v veliko pomoč, saj sva z njo pretvarjala vnešena števila v poljubnem številskem sistemu v bitna polja, jih ovrednotila, ter nazadnje pretvorila nazaj v prvoten številski sistem. Ker funkcija `pretvori` privzeto izpisuje na zaslon, sva ji dodala četrti argument, ki v primeru nastavitve `False` ne izpiše ničesar, pač pa le izračuna zeleno vrednost.

Tudi ta del aplikacije sva testirala z določenim spletnim orodjem.[8]

## 5 Branje iz datoteke

Zadnja točka je bila branje datoteke z računi in jih rešiti. Imamo neko datoteko z računi na disku, uporabnik pa to datoteko odpre v programu, in program mu izpiše rezultate računov. Za delo z datotekami v Pythonu sva morala dodati modul `filedialog` iz knjižnice Tkinter, s katerim sva lahko odpirala, zapirala ter brala iz datotek. Najprej sva si morala v kalkulatorju oz. v programu narediti novi zavihek Branje iz datotek. To sva naredila zato, da je program preglednejši in je posledično z njim lažje delati. Ko sva naredila svoj zavihek za delo z datotekami, sva v zavihku dodala zaslon, na katerega se izpiše vsebina datoteke ter rezultati, gumb s katerim lahko izbrava datoteko z diska, in še gumb, ki počisti zaslon. Ko uporabnik klikne na gumb za odpiranje datoteke, se kliče funkcija `odpri_txt`. Funkcija nato odpre raziskovalca (ang. File Explorer) v operacijskem sistemu Windows, kjer si lahko uporabnik izbere, iz katere datoteke bi rad prebral račune. Omejila sva tudi datoteke, da so lahko samo tipa `.txt`, tj., da so tekstovne. Ko uporabnik izbere svojo datoteko, začne funkcija<sup>[9]</sup> brati posamezne vrstice v datoteki in izpisovat rezultate. Ko pridemo do zadnje vrstice, v ozadju zapremo datoteko.

## Literatura

- [1] *Python eval error suppression*. <https://stackoverflow.com/questions/2140614/python-eval-error-suppression>. (12. 1. 2022).
- [2] *Right to left text in tkinter*. <https://stackoverflow.com/questions/20306726/right-to-left-text-in-tkinter>. (6. 1. 2022).
- [3] Abhilasha Chougule. *Do While Loop in Python*. <https://www.educba.com/do-while-loop-in-python/>. (30. 12. 2021).
- [4] *Split string into array in Python*. <https://stackoverflow.com/questions/44384854/split-string-into-array-in-python>. (31. 12. 2021).
- [5] *Prepend a string in Python*. <https://stackoverflow.com/questions/44792399/prepend-a-string-in-python>. (31. 12. 2021).
- [6] *Python Global Variables*. [https://www.w3schools.com/python/gloss\\_python\\_global\\_variables.asp](https://www.w3schools.com/python/gloss_python_global_variables.asp). (15. 1. 2022).
- [7] *RapidTables Base Converter*. <https://www.rapidtables.com/convert/number/base-converter.html>. (15. 1. 2022).
- [8] *Logical operations calculator*. <http://ambrsoft.com/MathCalc/LogicCalc/LogicOperations.htm>. (15. 1. 2022).
- [9] *Read a file line by line in Python*. <https://www.geeksforgeeks.org/read-a-file-line-by-line-in-python/>. (6. 1. 2022).