

**Kalkulator**

|  |  |
| --- | --- |
| Študent(-ka): | Matija PAJENK |
| Študent(-ka): | Rok ROZMAN MARINČIČ |
| Študijski program: | RIT 1 UN |
|  |  |
| Mentor(-ica): | Doc. dr. Petra Povalej Bržan, prof. rač. z. mat. |

Maribor, januar 2023

Kazalo

[1 Uvod 1](#_Toc124421961)

[2 Načrtovanje 2](#_Toc124421962)

[2.1 Psevdokoda 2](#_Toc124421963)

[2.2 Diagram poteka 3](#_Toc124421964)

[3 HTML 4](#_Toc124421965)

[3.1 Načrt 4](#_Toc124421966)

[3.2 Elementi 4](#_Toc124421967)

[4 CSS 5](#_Toc124421968)

[4.1 Oblika 5](#_Toc124421969)

[4.2 Prilagodljivost 5](#_Toc124421970)

[5 JavaScript 6](#_Toc124421971)

[5.1 Računske operacije 6](#_Toc124421972)

[5.2 Prednost operacij in oklepaji 7](#_Toc124421973)

[5.3 Vpis celotnega računa 7](#_Toc124421974)

[5.4 Pretvarjanje števil med številskimi sistemi 8](#_Toc124421975)

[5.5 Uporaba logičnih vrat 9](#_Toc124421976)

[5.6 Branje iz datoteke 9](#_Toc124421977)

[6 Viri 11](#_Toc124421978)

[Priloge 12](#_Toc124421979)

# Uvod

Naloga opisuje zgradbo in delovanje programa, ki je zmožen računati števila, pretvarjati med številskimi sistemi ter izvajati logične operacije. Program je v obliki spletne aplikacije, kjer zgradbo določa HTML, obliko CSS in logiko JavaScript.

# Načrtovanje

## Psevdokoda

*Izberemo vrsto računala preko radio gumba*

*Ob pritisku gumba z oznako "Vnos preko Tipkovnice" omogočimo vnos preko tipkovnice, privzeto izklopljeno*

*če izberemo Aritmetično:*

*če vnašamo račune ročno*

*vnesemo poljubno dolg račun v polje za računanje, a upoštevamo pravilnost zapisa*

*sicer*

*račune naložimo preko gumba "Izberi datoteko", ki sprejema tekstovne datoteke*

*iz izbrane datoteke prebere račune v obliki niza*

*nize doda v prvi stolpec tabele*

*v drugem stolpcu tabele je gumb "vstavi", ki vstavi izbrani račun v polje za računanje*

*ko imamo v polju za računanje željen račun, kliknemo gumb '='*

*vredonst polja se shrani v niz*

*če niz vsebuje oklepaje*

*rešimo vrednost znotraj oklepajev*

*v nizu zamenjamo podniz z oklepaji z rešeno vrednostjo*

*postopek ponovimo dokler imamo oklepaje*

*če niz vsebuje korene*

*podniz z korenskim zapisom zamenjamo z zapisu v obliki eksponenta - x^(1/2)*

*postopek ponovimo dokler imamo korene*

*če niz vsebuje eksponente*

*rešimo vrednost ekponeta*

*v nizu zamenjamo podniz eksponeta z rešeno vrednostjo*

*postopek ponovimo dokler imamo eksponente*

*če niz vsebuje množenje ali deljenje*

*dokler se v nizu pojavi množenje ali deljenje*

*preverimo, katera operacija se pojavi prej*

*če se množenje*

*rešimo vrednost množenja*

*v nizu zamenjamo podniz zmnožka z rešeno vrednostjo*

*če se deljenje*

*rešimo vrednost deljenja*

*v nizu zamnejamo podniz deljenja z rešeno vrednostjo*

*če niz vsebuje seštevanje ali odštevanje*

*rešimo vrednosti preostalega računa*

*končni rezultat izpišemo v polju za računanje*

*Sicer izberemo Logično:*

*zgorniji gumbi z oznakami 'BIN', 'OCT', 'DEC' in 'HEX' predstavljajo številski sistem za vhodne podatke,*

*spodnji gumbi z oznakami 'BIN', 'OCT', 'DEC' in 'HEX' pa predstavljajo številski sistem za izhodne podatke,*

*če vnasamo izjave ročno*

*vnesemo poljubno dolgo izjavo v polje za izjave, vendar upoštevamo pravilnost zapisa*

*sicer*

*izjave naložimo preko gumba "Izberi datoteko", ki sprejema tekstovne datoteke*

*iz izbrane datoteke prebere izjave v obliki niza*

*nize doda v prvi stolpec tabele*

*v drugem stolpcu tabele je gumb "vstavi", ki vstavi izbrano izjavo v polje za izjave*

*ko imamo v polju željeno izjavo, kliknemo gumb '='*

*vredonst polja se shrani v niz*

*če niz vsebuje logične operacije*

*prvo rešimo logični NOT*

*nato rešimo logični AND*

*nato rešimo logični OR in XOR v vrstnem redu kot se pojavita v računu*

*niz se pretvori iz vhodnega številskega sistema v izhodnega*

## Diagram poteka

Delovanje programa lahko enostavnejše prikažemo na diagramu poteka. Slika 1

A picture containing text, screen, electronics, black

Description automatically generated

Slika 1: Diagram poteka

# HTML

## Načrt

HTML določa zgradbo programa, kar pomeni, da večina elemenov, ki so vidni pri uporabi programa so določeni z html. Izjema so elementi, ki se tvorijo ob nalaganju tekstovnih datotek. V izvorni kodi, so vsi elementi zapisani eden za drugim, torej si lahko že ob pogledu kode predstavljamo izgled.

## Elementi

Za program so uporabljeni različni HTML elementi. Ogrodje kalkulatorja so različni <div> elementi, ki so neoblikovani blokovni elementi. Za vnos podatkov preko gumbov se uporablja element <input type="buuton">, ki ima vsaka svojo vrednost. Podatki naloženi preko datoteke so prikazani v tabeli.

# CSS

## Oblika

CSS skrbi za celotno obliko elementov in razmerja med njimi. To pomeni, da CSS določa velikost, barvo, pisavo elementov ter razmerje odmikov do drugih elementov. Za pisavo se uporablja pisava Lato, katera je vključena v CSS datoteko preko linka.

Za lažje vodoravno razporevanje elementov se uporablja flex. Flex lahko zasledimo na celotni apllikaciji, saj deli okno na dva dela, levo je kalkulator, desno pa je tabela z naloženimi računi. Poleg tega je še uporabljen znotraj kalkulatorja za razporeditev gumbov.

Poleg tega CSS še omogoča vse animacije, ki se izvajajo ob uporabi programa.

Gumbi tipa "radio" in "checkbox" imajo tudi obliko po meri. [1]

## Prilagodljivost

Aplikacija je prilagodliva glede na napravo oz. velikost zaslona na kateri se izvaja [2]. Pri dovolj velikem zaslonu bodo elementi razporejeni vodoravno, torej kalkulator levo in tabela desno. Pri manjših zaslonih to ni možno, zato se elemetni razporedijo eden nad drugega, torej kalkulatorj na vrhu in tabela pod njim. Slika 2

Text

Description automatically generated

Slika 2: CSS za prilagodlivost glede na velikost zaslona

# JavaScript

Izbira za JavaScript je sledila iz tega, da sem jezik uporabljal že v preteklosti in ker je zelo močan in netipiziran jezik, kar olajša pisanje kode, kjer se pojavlja večkratno pretvarjanje med nizi in števili. [3]

## Računske operacije

Implementacija operacij ni zahtevna, saj jih programski jezik JavaScript že podpira, torej ne potrebujemo ustvarjani novih algoritmov, temveč uporabimo že obstoječe. Program podpira naslednje računske operacije:

* Seštevanje (+),
* Odštevanje (-),
* Množenje (\*),
* Deljenje (/),
* Mod – ostanek pri deljenju (%) ter
* Potenciranje – eksponenti in koreni (\*\*).

V oklepajih so operatorji, ki se uporabljajo v JavaScript-u. Operacije dodamo preko gumbov v grafičnem vmesniku. Slika 3

Graphical user interface, application

Description automatically generated

Slika 3: Grafični vmesnik – aritmetično računalo

## Prednost operacij in oklepaji

Kalkulator mora za pravilno računanje upoštevati tudi prednost operacij, ki si sledijo tako:

* Oklepaji,
* Potenciranje,
* Množenje, deljenje in mod ter
* Seštevanje in odštevanje.

## Vpis celotnega računa

Kalkulator podpira reševanje celotnih računov. To je mogoče, saj račun znotraj polja obravnava kot niz. Znotraj niza upošteva prednost operacij in postopoma rešuje vrednosti znotraj podnizev ter to vrednost zamenja z izrazu v prvotnem nizu. Ta postopek ponovi dokler niz ni enak neki številki, ki predstavlja vrednost celotnega računa. [4], [5]

Primer:

5 \* (3 + 7) : Pri tem tem računu program prvo reši vrednost znotraj okepajev, torej 3 + 7 = 10, nato rešeno vrednost zanmenja v prvotnem računu in dobimo nov račun, ki je oblike: 5 \* 10, preverimo, če je vrednost število, končamo, kar ni v tem primeru, torej rešujemo naprej. Program ovrednoti novi račun 5 \* 10 = 50 in zamenja vrednost znotraj prejšnega računa. Tako postane vrednost računa 50. Ko je vredonst računa število kočamo računanje in izpišemo dobljeno vrednost v oknu, kjer smo prvotno vpisali račun.

## Pretvarjanje števil med številskimi sistemi

Pretvarjanje med števili je mogoče preko logičnega računala. Omogča pretvarjanje med naslednjimi številskimi sistemi:

* Dvojiško (BIN),
* Osmiško (OCT),
* Desetiško (DEC) ter
* Šestnajstiško (HEX).

Pretvarjanje je možno med poljubnima dvema številskima sistema. Ob kliku na enačaj se niz znotraj polja direktno pretvori iz vhodnega v izhodni številski sistem. Zgornja vrstica predstavlja vhodni številski sistem, medtem ko spodnja izhodnega. Hkrati pa so omogočeni le določeni gumbi za vnos vrednosti, torej tisti katere številski sistem podpira. Slika 4

Graphical user interface, application

Description automatically generated

Slika 4: Grafični vmesnik – logično računalo

## Uporaba logičnih vrat

Poleg pretvarjanja logično računalo podpira še vrednotenje logičnih operacij:

* NOT,
* AND,
* OR ter
* XOR.

Program deluje, tako da prvo reši vrednost logične izjave, nato pa pretvori v izbran številski sistem.

Primer: Dvojiško izjavo 101 AND 100 želimo pretvoriti v desetiško vrednost. Prvo se ovrednoti vredost izjave, torej dvojiško 100, nato se pretvori v izbran številski sistem. Končni rezultat je torej desetiško 4.

## Branje iz datoteke

Oba računala podpirata nalaganje računov iz tekstovne datoteke, pri tem je potrebno upoštevati primerno obliko zapisa znotraj tekstovnih datotek. Za aritmetično računalo so posamezni računi v svoji vrstici. Zapis za logično računalo pa je oblike :

"vhodni številski sistem; izraz; izhodni številski sistem"

Za delovanje je uporabljen element tipa <input type="file"> [6], ki pokliče funkcijo, ki uporablja objekt tipa FileReader(), ki omogoča branje lokalnih datotek [7].

Program vsebino datoteke loči v polje nizov, kjer vsaka vrstica v datoteki predstavlja svoj niz. Pri aritmetičnem računalu se nizi dodajo v tabelo, ki ima tudi gumb "vstavi", kateri vstavi račun v računalo. Slika 5

Chart

Description automatically generated with low confidence

Slika 5: Tabela računov za aritmetično računalo

Pri logičnem računalo je še dodana funkcionalnost, da izbere vhodni in izhodni številski sistem. Slika 6

Chart

Description automatically generated

Slika 6: Tabela računov za logično računalo

# Viri

[1] “How To Create a Custom Checkbox and Radio Buttons.” https://www.w3schools.com/howto/howto\_css\_custom\_checkbox.asp (accessed Jan. 04, 2023).

[2] “CSS @media Rule.” https://www.w3schools.com/cssref/css3\_pr\_mediaquery.php (accessed Dec. 22, 2022).

[3] “JavaScript Tutorial.” https://www.w3schools.com/js/ (accessed Dec. 22, 2022).

[4] “Insert String at specific Index of another String in JS | bobbyhadz.” https://bobbyhadz.com/blog/javascript-insert-string-at-index (accessed Dec. 22, 2022).

[5] “Remove a Substring from a String using JavaScript | bobbyhadz.” https://bobbyhadz.com/blog/javascript-remove-substring-from-string (accessed Dec. 22, 2022).

[6] “<input type="file"> - HTML: HyperText Markup Language | MDN.” https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/file (accessed Dec. 22, 2022).

[7] “File and FileReader.” https://javascript.info/file (accessed Dec. 22, 2022).

# Priloge



