

David Rajlič, Andraž Šošterič

**Kalkulator**

Seminarska naloga 2

Maribor, januar 2023

# Kazalo vsebine

[**Kazalo vsebine**](#_mprydvmydr54) **2**

[**1 Uvod**](#_du97dpgv0rpi) **3**

[**2 Implementirane Funkcije**](#_j7urqewv2kwx) **4**

[**3 Reševanje datotek**](#_2pjypidk3q8g) **6**

[3. 1 Uporaba grafičnega vmesnika](#_vh42e2e9eq6k) 6

[3. 2 Podprti tipi datotek:](#_byala1px8tkl) 6

[3. 3 Psevdokod:](#_gnyfzqnx3ijg) 6

[3. 4 Diagram poteka](#_teee3jg0kcol) 7

[**4 Računanje računskih operacij**](#_78m6u8nrze5z) **8**

[4. 1 Psevdokoda za reševanje uporabnikove vnešene enačbe v obliki besedila](#_xoohe4q7gjb6) 8

[**5 Pretvarjanje številskih sistemov**](#_y8xgwqyc2w7j) **9**

[5. 1 Uporaba grafičnega vmesnika:](#_czh54vvb5h4q) 9

[5. 2 Format v datoteki:](#_jft7l13z3uxe) 9

[5. 3 Slovar standardnih oznak sistemov:](#_dqkxhhwld2fg) 9

[5. 4 Poljubni številski sistem:](#_ie2b2whm4b7b) 9

[5. 5 Psevdokod](#_c8g6mdomtw7o) 10

[5. 6 Diagram poteka](#_6qaex4wo6ngs) 11

[**6 Logične operacije**](#_njlgtjfy6kw1) **12**

[6. 1 Uporaba grafičnega vmesnika](#_17nxtnq4alvr) 12

[6. 2 Format v datoteki](#_gssb4oczw9af) 13

[6. 3 Podprta vrata](#_bdebc02n7uwn) 13

[6. 4 Številski sistemi](#_sgzv2y857uiq) 13

[6. 5 Psevdokod](#_czn9w410zopm) 14

[6.7 Diagram Poteka](#_pvcziyee6uou) 15

[**7 Viri**](#_776qeahhvmee) **16**

[7. 1 Dokumentacija](#_c5tx073vxd9o) 16

[7. 2 Viri slik](#_gu8vd5xu2j65) 16

# 

# 1 Uvod

Pri tej seminarski nalogi sva izdelala kalkulator, ki poleg osnovnih računskih operacij (seštevanje, odštevanje, množenje in deljenje), uspešno izračuna tudi bolj kompleksne operacije, kot so npr. korenjenje, potenciranje, pretvarjanje številskih sistemov itd.

Kalkulator je v celoti narejen v programskem jeziku C++. Grafični vmesnik je ustvarjen z .NET. Koda se nahaja v 7 datotekah, te so:

· MyForm.cpp: Vsebuje main() funkcijo, ki zažene program

· MyForm.h: Vsebuje izgled aplikacije in funkcionalnost gumbov in drugih elementov

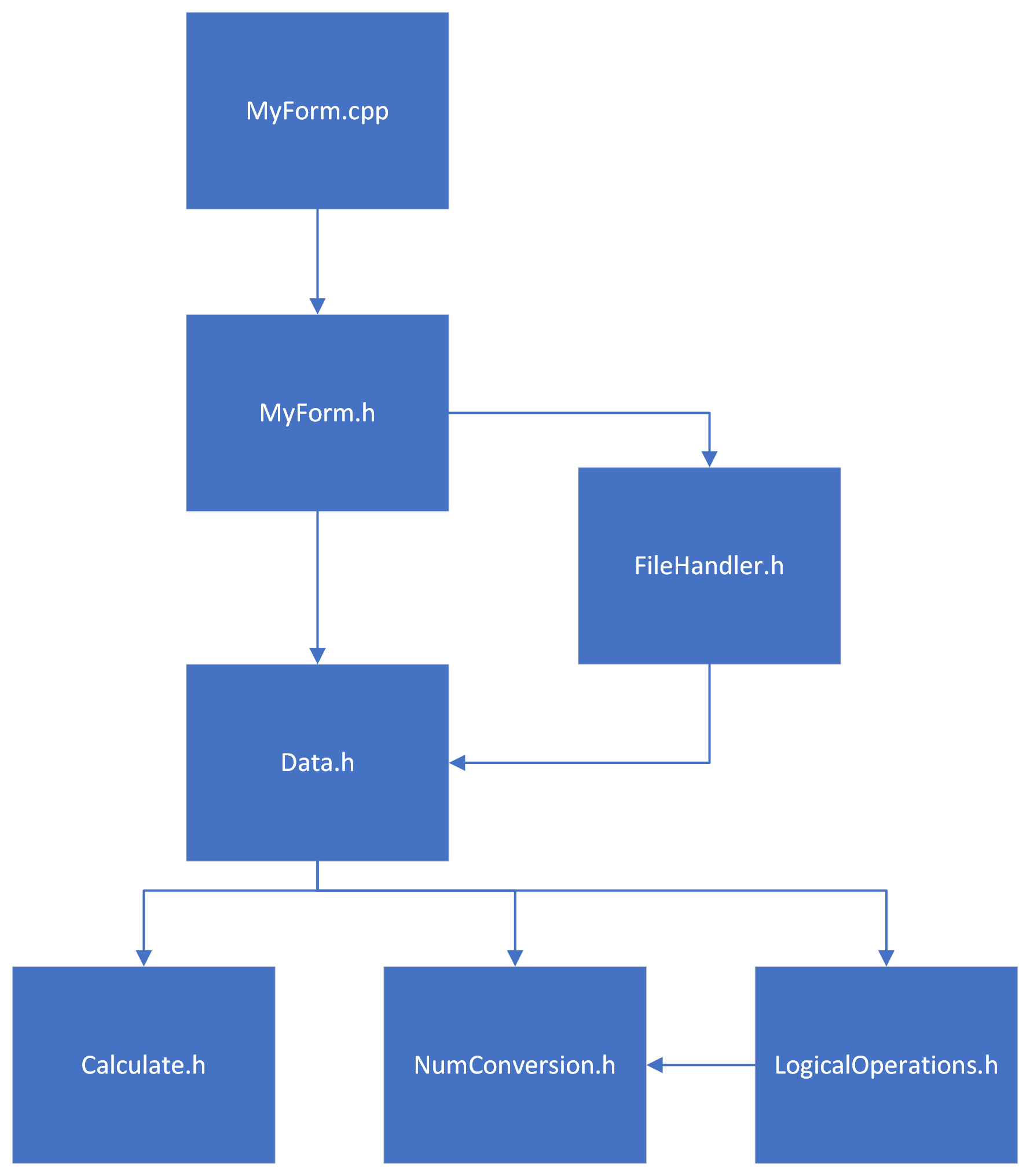
· Data.h: vsebuje funkcije in spremenljivke, ki povezujejo MyForm.h z funkcijami v drugih datotekah, ki željene račune izvedejo

· FileHandler.h: Poskrbi za branje iz datotek, reševanje prebranega in izpis rezultatov, tako kot MyForm.h kliče funkcije iz Data.h

· NumConversion.h: poskrbi za pretvarjanje med številskimi sistemi

· LogicalOperations.h: poskrbi za logične operacije

· Calculate.h: Poskrbi za klasične račune

Klici funkcij:

# 

# 

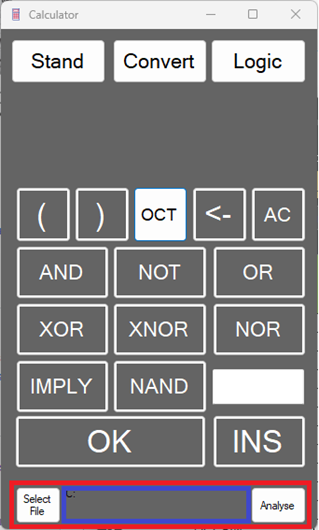
# 2 Implementirane Funkcije

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Funkcija/vsebina | Implementiral (ime člana skupine) | Lastna implementacija (D/N) | Uporaba knjižnice (D/N) | Format uporabe |
| + | David Rajlič | D | N | a+b |
| - | David Rajlič | D | N | a-b |
| \* | David Rajlič | D | N | a\*b |
| / | David Rajlič | D | N | a / b |
| modul | David Rajlič | D | N | a % b |
| potence | David Rajlič | D | N | a ^ b (GUI) in pow(a,b) |
| koren | David Rajlič | D | N | sqrt(a) |
| vpis računa | David Rajlič | D | N | npr. (a+b) \* sqrt(c) |
| BIN-OCT | Andraž Šošterič | D | N | BIN a OCT |
| BIN-DEC | Andraž Šošterič | D | N | BIN a DEC |
| BIN-HEX | Andraž Šošterič | D | N | BIN a HEX |
| OCT-BIN | Andraž Šošterič | D | N | OCT a BIN |
| OCT-DEC | Andraž Šošterič | D | N | OCT a DEC |
| OCT-HEX | Andraž Šošterič | D | N | OCT a HEX |
| DEC-BIN | Andraž Šošterič | D | N | DEC a BIN |
| DEC-OCT | Andraž Šošterič | D | N | DEC a OCT |
| DEC-HEX | Andraž Šošterič | D | N | DEC a HEX |
| HEX-BIN | Andraž Šošterič | D | N | HEX a BIN |
| HEX-OCT | Andraž Šošterič | D | N | HEX a OCT |
| HEX-DEC | Andraž Šošterič | D | N | HEX a DEC |
| direktno pretvarjane | Andraž Šošterič | D | N | nnn a nnn  (gl. pog. 5. 2) |
| logična vrata na BIN | Andraž Šošterič | D | N | a nnn a …  (gl. pog. 6. 2) |
| logična vrata na poljubnem sist. | Andraž Šošterič | D | N | xxx a nnn a  (gl. pog. 6. 3) |
| branje/zapis datoteke MAT | Andraž Šošterič | D | N | Klik na gumb  (gl. pog 4. 1) |
| branje/zapis datoteke št.sistemi | Andraž Šošterič | D | N | Klik na gumb  (gl. pog 4. 1) |
| branje/zapis datoteke log.vrata | Andraž Šošterič | D | N | Klik na gumb  (gl. pog 4. 1) |

# 

# 3 Reševanje datotek

## 3. 1 Uporaba grafičnega vmesnika

Za delo z datotekami uporabimo del obrobljen z rdečo.

· Select file: Odpre raziskovalca (File Explorer) v katerem izberemo eno datoteko tipa .txt

· Modro polje: tukaj se pojavi pot do datoteke

· Analyse: S klikom na ta gumb sprožimo reševanje datoteke

· **OPOMBA:** Imena datotek, in zavihkov vsebovanih v poti ne smejo vsebovati šumnikov.

## 3. 2 Podprti tipi datotek:

1. Standardni računi

2. Pretvorbe med številskimi sistemi

3. Logične operacije

4. Logične operacije nad poljubnim številskim sistemom

5. Mešanice računov v točkah od 1 do 4

Formati zapisa v datotekah so opisani v pripadajočih poglavjih.

## 3. 3 Psevdokod:

Ustvari in odpri nov FileDialog

Od File Dialoga pridobi pot

Zapri FileDialog

Prikaži pot na zaslonu

Ob kliku na Analyse odpri datoteko na podani poti

Če je odprta jo rešuj vrstico po vrstico in rezultate shranjuj v vektor

Iz vektorja v datoteko zapiši rezultate

Zapri datoteko

## 3. 4 Diagram poteka

## 

## 

# 4 Računanje računskih operacij

Program rešuje osnovne računske operacije, kot so npr. seštevanje, odštevanje, množenje in deljenje ter tudi kompleksnejše računske operacije. Te osnovne računske operacije sem razdelil na posamezne funkcije, poleg tega sem ustvaril tudi posebej funkcijo za računanje ostanka oz.modul, za korenjenje in potenciranje. Na koncu sem pa še ustvaril funkcijo, ki sprejme enačbo, ki jo uporabnik vnese v obliki besedila, funckija to enačbo sprejme in jo pravilno reši.

## 

## 

## 

## 

## 4. 1 Psevdokoda za reševanje uporabnikove vnešene enačbe v obliki besedila

1. uporabniku daj možnost vstavljanja enačbe
2. uporabnikovo vnešeno enačbo sprejmi in podaj v funkcijo kot parameter
3. pokliči funkcijo, ki bo to enačbo rešila
4. pojdi čez vsako črko oz.element v besedilni enačbi in preveri za kakšen element gre (za črko, številko, presledek)
5. če je številka si jo shrani in zapomni
6. če je črka in naslednje črke tvorijo besedo (sqrt ali pow) si tudi zapomni in izvedi njune funkcije
7. če je pa presledek pa ignoriraj in se prestavi za eno naprej
8. če znak +, -, \*,/ si tudi zapomni in shrani to računsko operacijo
9. sproti računaj operacije med seboj, uporabi števila in računsko operacijo, ki si si jih shranil in zapomnil
10. če je računska operacija v oklepaju, jo prvo reši, šele nato ostale operacije oz. preostali del enačbe

# 5 Pretvarjanje številskih sistemov

## 5. 1 Uporaba grafičnega vmesnika:

· Rdeče: Sistem iz katerega pretvarjamo

· Modro: Sistem v katerega pretvarjamo

· Zeleno: Vrednost, ki jo želimo pretvoriti

· AC: Počisti vse vrednosti in izbire

· Convert: potrdi izbiro in izvede pretvorbo

## 5. 2 Format v datoteki:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Primer : | BIN\_ | 111 | \_DEC\_ | = |
| Opis: | Iz sistema | Število | V sistem | Je enako |

\_ - predstavljajo presledke, vsakega izmed njih naj uporabnik v datoteki zamenja z natanko enim presledkom.

**OPOMBA**: Za sisteme višje od desetiškega naj uporabnik uporablja le velike tiskane črke (npr. HEX 9AE DEC = 2478 in ne HEX 9ae DEC = Number error digit out of bounds, A = 10, a = 36, ni podprto)

## 5. 3 Slovar standardnih oznak sistemov:

· BIN – Dvojiški

· OCT – Osmiški

· DEC – Desetiški

· HEX – Šestnajstiški

## 5. 4 Poljubni številski sistem:

· Podpora številskih sistemov med osnovo 2 in 35.

· Datoteke: CNN - pri čemer je NN dvomestna desetiška oznaka (Npr. BIN = C02, DEC = C10, petiški = C05)

· Grafični vmesnik: Napišemo samo osnovo številskega sistema (Npr. BIN = 2, DEC = 10, petiški = 5)

· Za pomembnejše sisteme lahko uporabljamo standardne oznake, ali pa poljubnega

## 5. 5 Psevdokod

Iz prvih in zadnjih treh znakov vnosa dobi podatke o sistemu iz katerega želimo pretvarjati in v katerega želimo pretvarjati

Iz vmesnih znakov pridobi podatek o številki, ki jo želimo pretvoriti

Preveri če so vnešeni sistemi veljavni

Če je vse veljavno začne pretvorbo,

Preveri če je kateri izmed sistemov desetiški, če je izvede pretvorbo

Če ni izvede pretvorbo v desetiškega in nato v željenega direktno, brez uporabnikovega vnosa.

Za pretvorbo iz desetiškega sistema program deli število z deset in si zapomni ostanek,

Če je ostanek manjši od deset v nov niz shrani ascii znak za ostanek + 48,

če je pa večji pa shrani ascii znak za ostanek + 55.

Na koncu program obrne niz, da dobi pravo številko.

Za pretvorbo v desetiški sistem program najprej obrne niz,

Definiramo int sum

Za vsak znak v podanem nizu

Če je ascii koda za trenutni znak manjša od 58, v vrednost mod shrani 48

Drugače pa v vrednost mod shrani 55

vrednosti sum prišteje (ascii vrednost - mod) \* (dani številski sistem)^(vrednost ponovitve zanke)

Na koncu int sum spremeni v tip niz in ga obrne.

## 

## 5. 6 Diagram poteka

# 6 Logične operacije

## 6. 1 Uporaba grafičnega vmesnika

· Rdeče: Sistem znotraj katerega izvajamo operacije, zamenjava ob kliku na gumb

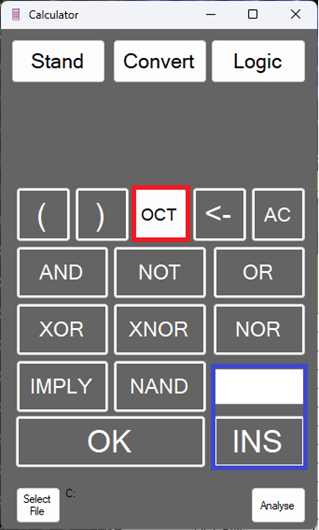
· Modro: Logične operacije

· Zeleno: Vnos vrednosti

· AC: Počisti vse vrednosti in izbire

· <-: Počisti zadnji simbol

· OK: potrdi izbiro in izvede logične operacije



· Rdeče: Sistem znotraj katerega izvajamo operacije, zamenjava ob kliku na gumb

· Modro: Vstavljanje poljubnega števila, število vnesemo v vnosno polje znotraj modrega okvirja in kliknemo gumb ins.

## 6. 2 Format v datoteki

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Primer : | BIN | 1 | OR | 0 | … | = |
| Opis: | Iz sistema  (Ni obvezno) | Število | Logična vrata | Število | Poljubno število operacij | Je enako |

Podprti tudi oklepaji.

**POMEMBNO: MED vsakim elementom mora biti NATANKO en presledek**

## 6. 3 Podprta vrata

· NOT – negacija (ne)

· AND – konjunkcija (IN)

· OR – disjunkcija (ALI)

· XOR – ekskluzivni ali (ALI a ALI b)

· XNOR – ekvivalenca (ČE in samo ČE)

· NOR – (NE ALI)

· IMPLY – implikacija

· NAND – (NE IN)

## 6. 4 Številski sistemi

· Če ni podan, program predpostavi, da gre za dvojiškega (BIN)

· V grafičnem vmesniku podprti sistemi BIN, OCT, DEC, HEX

· V datotekah podprti sistemi od BIN do osnove 35, podajo se tako kot v pretvarjanju številskih sistemov (S standardnimi oznakami ali CNN)

## 

## 6. 5 Psevdokod

Ali niz vsebuje oznako za številski sistem

Če jo ga pretvori v željen številski sistem

Ali vsebuje oklepaje

Če jih poišči naslednji oklepaj in shrani njegov indeks

Če najde zaklepaj shrani podniz od indeksa do zaklepaja in ga reši

Preveri če je konec

Izberi trenutni podniz za i in preveri, če ustreza logični operaciji

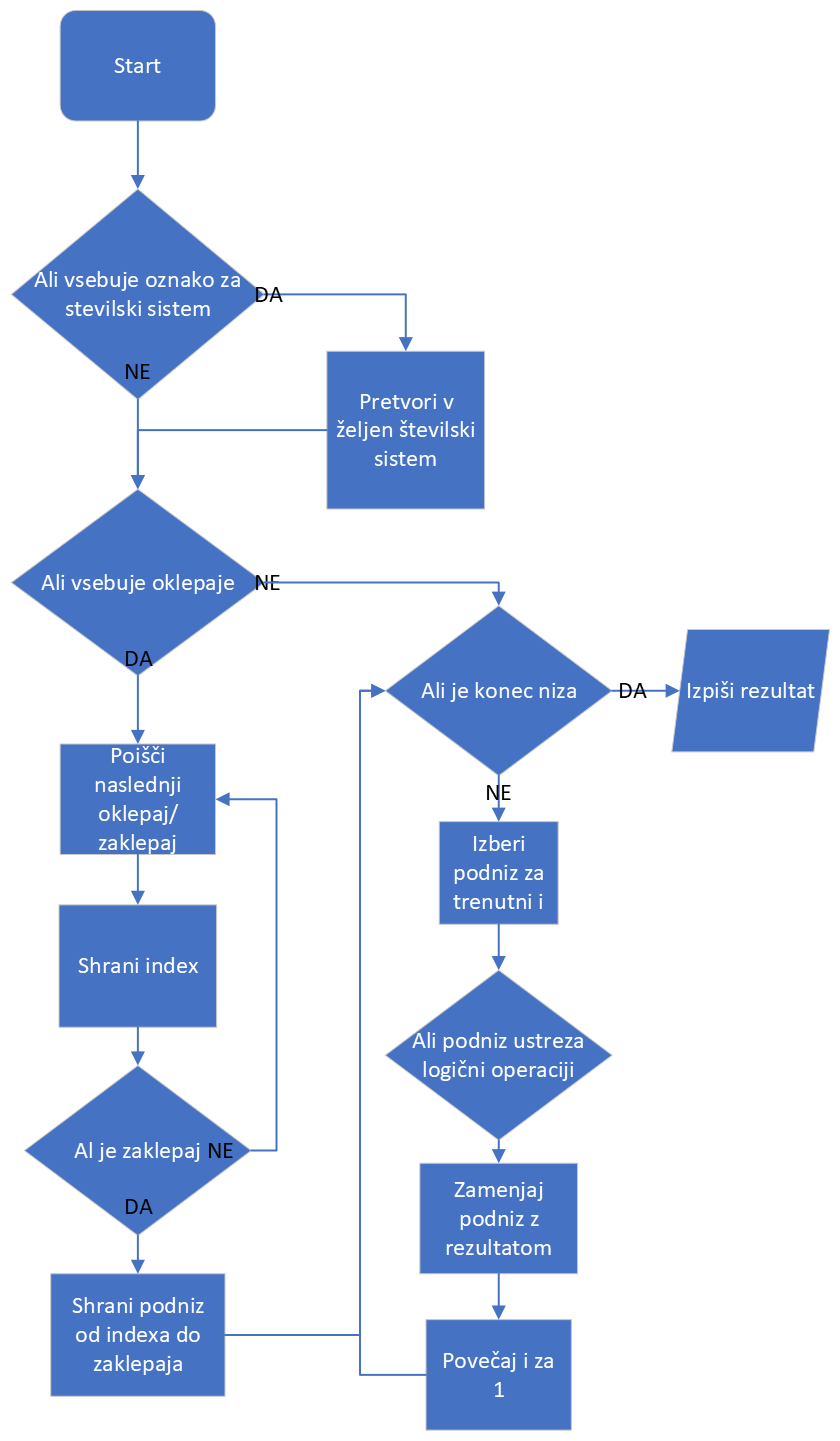
Če ustreza zamenjaj podniz z rezultatom

Povečaj i za 1

Na koncu izpiši rezultat

Če oklepajev ni preskoči iskanje oklepajev

## 6.7 Diagram Poteka



# 7 Viri

## 7. 1 Dokumentacija

1 https://learn.microsoft.com/en-us/dotnet/ (Zadnji dostop: 13. 1. 2023)

2 https://learn.microsoft.com/en-us/cpp/dotnet/ (Zadnji dostop: 14. 1. 2023)

3 https://learn.microsoft.com/en-us/cpp/cpp/header-files-cpp?view=msvc-170(Zadnji dostop: 20. 12. 2022)

4 https://cplusplus.com/reference/vector/vector/ <https://learn.microsoft.com/en-us/cpp/dotnet/> (Zadnji dostop: 6. 1. 2023)

## 7. 2 Viri slik

Ikona kalkulatorja:

https://iconarchive.com/show/android-style-honeycomb-icons-by-wwalczyszyn/Calculator-icon.html (Dostop: 14. 1. 2023) Avtor**:** Wallec Licenca: https://creativecommons.org/licenses/by-nc-nd/4.0/