Makro  
  
Sub Zacetek

PomnozCelaStevila

Dim ObravnavanDokument As Object

ObravnavanDokument = ThisComponent

Dim DokumentnoBesedilo As Object

DokumentnoBesedilo = ObravnavanDokument.getText()

Dim SkupnoSoglasnikov As Long

SkupnoSoglasnikov = IzracunajSoglasnike(DokumentnoBesedilo.getString())

MsgBox "Število soglasnikov: " & SkupnoSoglasnikov

End Sub

Sub PomnozCelaStevila

Dim BesedilniKazalec As Object

BesedilniKazalec = ThisComponent.Text.CreateTextCursor()

BesedilniKazalec.gotoStart(False)

Do

BesedilniKazalec.goToEndOfWord(True)

Dim TrenutnoBesedilo As String

TrenutnoBesedilo = BesedilniKazalec.String

If IsNumeric(TrenutnoBesedilo) And Fix(Val(TrenutnoBesedilo)) = Val(TrenutnoBesedilo) Then

Dim DolgoStevilo As Long

DolgoStevilo = CLng(TrenutnoBesedilo)

Dim ZmnozenoBesedilo As String

ZmnozenoBesedilo = CStr(DolgoStevilo \* 4)

BesedilniKazalec.String = ZmnozenoBesedilo

End If

Loop While BesedilniKazalec.gotoNextWord(False)

End Sub

Function IzracunajSoglasnike(Tekst As String) As Long

Dim StevecSoglasnikov As Long

StevecSoglasnikov = 0

Dim Indeks As Integer

For Indeks = 1 To Len(Tekst)

Select Case Mid(Tekst, Indeks, 1)

Case "b", "c", "č", "d", "f", "g", "h", "j", "k", "l", "m", "n", "p", "r", "s", "š", "t", "v", "z", "ž", \_

"B", "C", "Č", "D", "F", "G", "H", "J", "K", "L", "M", "N", "P", "R", "S", "Š", "T", "V", "Z", "Ž"

StevecSoglasnikov = StevecSoglasnikov + 1

End Select

Next Indeks

IzracunajSoglasnike = StevecSoglasnikov

End Function

MAXIMA

ODVOD

f1: diff(sin(x) + log(x) + sqrt(x), x);

f2: diff(atan((2\*x + 1)^2 / x^2), x);

f3: diff(x^sin(x), x);

f4: diff(2^(x^(2\*x)), x);

f5: diff(cos(x^2), x);

LIMITI

f6: limit((1 - x^2) / sin(x), x, 1);

f7: limit(((n - 2) / n)^(n / 2), n, inf);

f8: limit(x \* (sqrt(x^2 + 1) - x), x, inf);

f9: limit(x^2 \* exp(-x), x, inf);

f10: limit(1 / sin(x) - 1 / tan(x), x, 0);

INTEGRALI

f11: integrate((x^3 - 2\*x + 6) / x, x);

f12: integrate(x^3 \* exp(x), x);

f13: integrate(x^2 \* sin(2\*x), x);

f14: integrate(sqrt(5\*x - 6), x);

f15: integrate(cos(x) \* sin(x)^6, x);

DOLOCENI INTEGRALI

f16: integrate(1 / (4 - x)^(2/3), x, 2, 6);

f17: integrate(1 / (2 \* sin(x)^2 + 1), x, 0, %pi / 4);

f18: integrate(sqrt(exp(x) - 1), x, 0, log(2));

f19: integrate(1 / sqrt(x^2 + 2\*x + 4), x, -2, 0);

f20: integrate(4 \* x^(1/3) - 2 / x^(1/3), x, 1, 8);

A math equations on a white background

Description automatically generatedA white background with black text and black numbers

Description automatically generated

Regex

1. Basic Concepts:

Literal Characters: The simplest form of regex. It matches the exact character or sequence of characters. For example, cat matches "cat" in "catapult".

Metacharacters: Special characters that have a unique meaning in regex. Some common metacharacters are . ^ $ \* + ? { } [ ] \ | ( ).

2. Common Metacharacters:

.: Matches any single character except newline characters. For example, h.t matches "hat", "hit", "hot", etc.

^: Matches the start of a string. ^A matches "A" in "Apple" but not in "Banana".

$: Matches the end of a string. end$ matches "end" in "weekend" but not in "ending".

\*: Matches the preceding element zero or more times. bo\* matches "booooo" or just "b".

+: Matches the preceding element one or more times. a+ matches "a" in "candy" and all a's in "caaaaaaandy".

?: Makes the preceding element optional. It matches zero or one occurrence. colou?r matches "color" and "colour".

[]: Matches any one of the characters inside the square brackets. [abc] matches "a", "b", or "c".

[^]: Matches any character not inside the square brackets. [^abc] matches any character except "a", "b", or "c".

{n}: Matches exactly n occurrences of the preceding element. a{3} matches "aaa".

{n,}: Matches n or more occurrences of the preceding element. a{2,} matches "aa", "aaa", "aaaa", etc.

{n,m}: Matches at least n and at most m occurrences of the preceding element. a{2,4} matches "aa", "aaa", or "aaaa".

\: Escapes a metacharacter. It turns special characters into literal characters. \. matches a period.

3. Special Sequences:

\d: Matches any digit. Equivalent to [0-9].

\D: Matches any non-digit. Equivalent to [^0-9].

\s: Matches any whitespace character.

\S: Matches any non-whitespace character.

\w: Matches any alphanumeric character (including underscore). Equivalent to [a-zA-Z0-9\_].

\W: Matches any non-alphanumeric character.

4. Grouping and Capturing:

(): Groups multiple tokens together and creates a capture group for extracting a substring or using a backreference.

5. Quantifiers:

Quantifiers like \*, +, and ? can be made lazy (non-greedy) by adding a ?. This makes them match as little as possible. .\*? is a non-greedy match for any character.

6. Flags:

Regex can be used with different flags, like case insensitive (i), multiline (m), dotall (s), etc., which change how the pattern is interpreted.

Examples:

/cat/: Matches "cat" in "catapult" or "concatenate".

/\d{2,4}/: Matches a sequence of 2 to 4 digits.

/^https?:\/\/\S+$/: Matches a simple HTTP or HTTPS URL.