



Marijan Dizdar

Smath Studio
-priručnik
za apsolutnog početnika

Novi Sad , 2011.

Sadržaj

Reč autora	1
Uvod	3
Izgled	5
Regioni	9
Tekstualni region	9
Unos matematičkih vrednosti.....	15
Rad sa varijablama (promenljive).....	17
Funkcije	19
Definisanje jedinica.....	20
Matrice i vektori	24
Rešavanje sistema linearnih jednačina	29
Opseg (Range)	30
Izvodi i integrali	31
Grafici	33
Rešavanje jednačina i nule funkcije.....	41
Pisanje i čitanje podataka iz fajla	45
Programiranje	47
IF komanda	48
FOR petlja	55
WHILE petlja.....	65
Spisak funkcija.....	68
Matrice i vektori.....	68
Kompleksni brojevi	71
Trigonometrija	72
Hiperbolični	73
Programiranje	75
Stringovi.....	75
Fajlovi.....	76

Reč autora

Tokom studiranja u više navrata sam koristio program Mathcad 14. Njegovom primenom dosta mi je bilo olakšano rešavanje određenih zadataka kao i prezentovanje istih. Nedavno sam na internetu pronašao program SMath Studio koji mi se učinio kao dobro alternativno rešenje. Posle kraćeg korišćenja odlučio sam da spojim "lepo i korisno", da napišem jedan mali priručnik i time nekom pomognem, a i da se na neki način zahvalim autoru samog programa na izuzetnom radu.

Sam priručnik je napisan za osobe koje nikad nisu koristili ni Mathcad, a ni SMath Studio 0.89 (build: 8).

Takođe bih se zahvalio Prof. Gilberto E. Urroz-u koji mi je dozvolio da iskoristim njegovo materijal prilikom izrade ovog priručnika.

Uvod

SMath Studio je matematički program koji po izgledu veoma podseća na Mathcad. Bitna razlika između ova dva programa je i to što je SMath besplatan dok je cena Mathcad-a (Mathcad Prime) nekoliko stotina evra. Naravno postoji još bezbroj razlika ali ih ovom prilikom neću spominjati.

Sam program je preveden na dosta jezika i radi na nekoliko platformi (Windows, Windows Mobile i Linux). SMath je delo ruskog programera po imenu Andraj Ivašov.

Glavna web stranica: <http://en.smath.info/forum/>

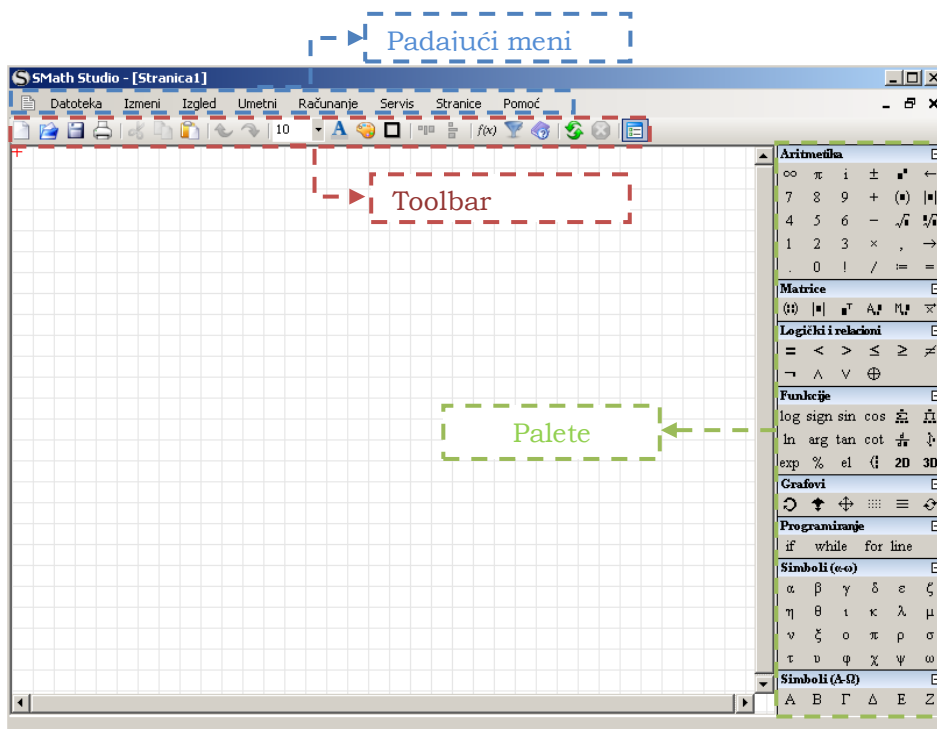
Postoji i live verzija program koja se može koristiti direktno iz browser-a i može se naći na adresi: <http://smath.info/live/?lang=eng>

Dostupna je i portable verzija programa za slučaj da na računaru koji koristite nemate administratorsko pravo da instalirate programe. Tako da ovu verziju možete koristiti sa USB Flash memorije.

Mislim da su ove informacije sasvim dovoljne za uvod tako da možemo početi da upoznajemo ovaj zanimljiv program.

Izgled

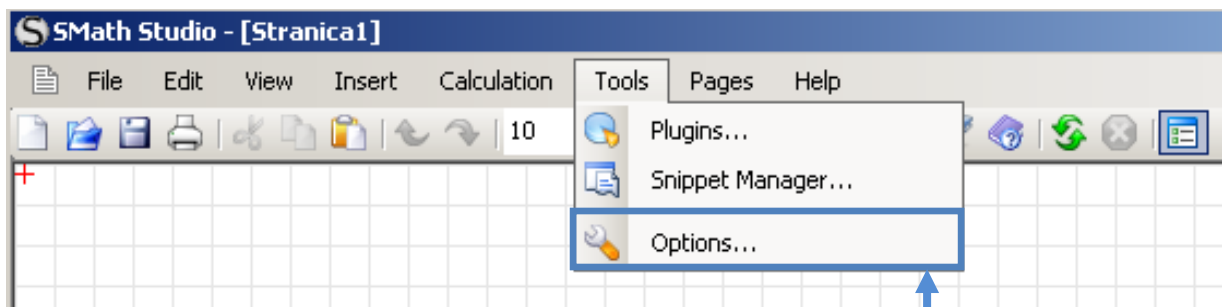
Prilikom pokretanja programa otvara nam se sledeći izgled radnog ekrana.



Slika 1. Izgled prozora programa SMath Studio

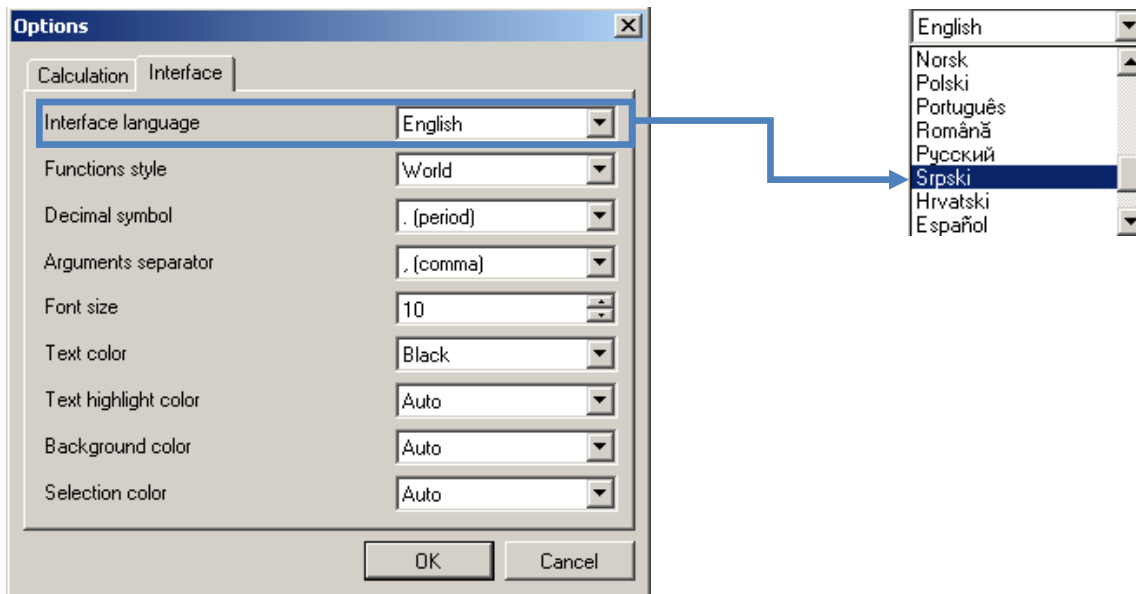
Primećujemo da na gornjoj strani postoje padajući meniji, toolbar, a sa desne nekoliko paleta.

Odmah na samom početku da prikazemo promenu jezika posle instalacije programa.



Slika 2. Promena jezika programa 1/2

Izborom iz padajućeg menija Tool/Options



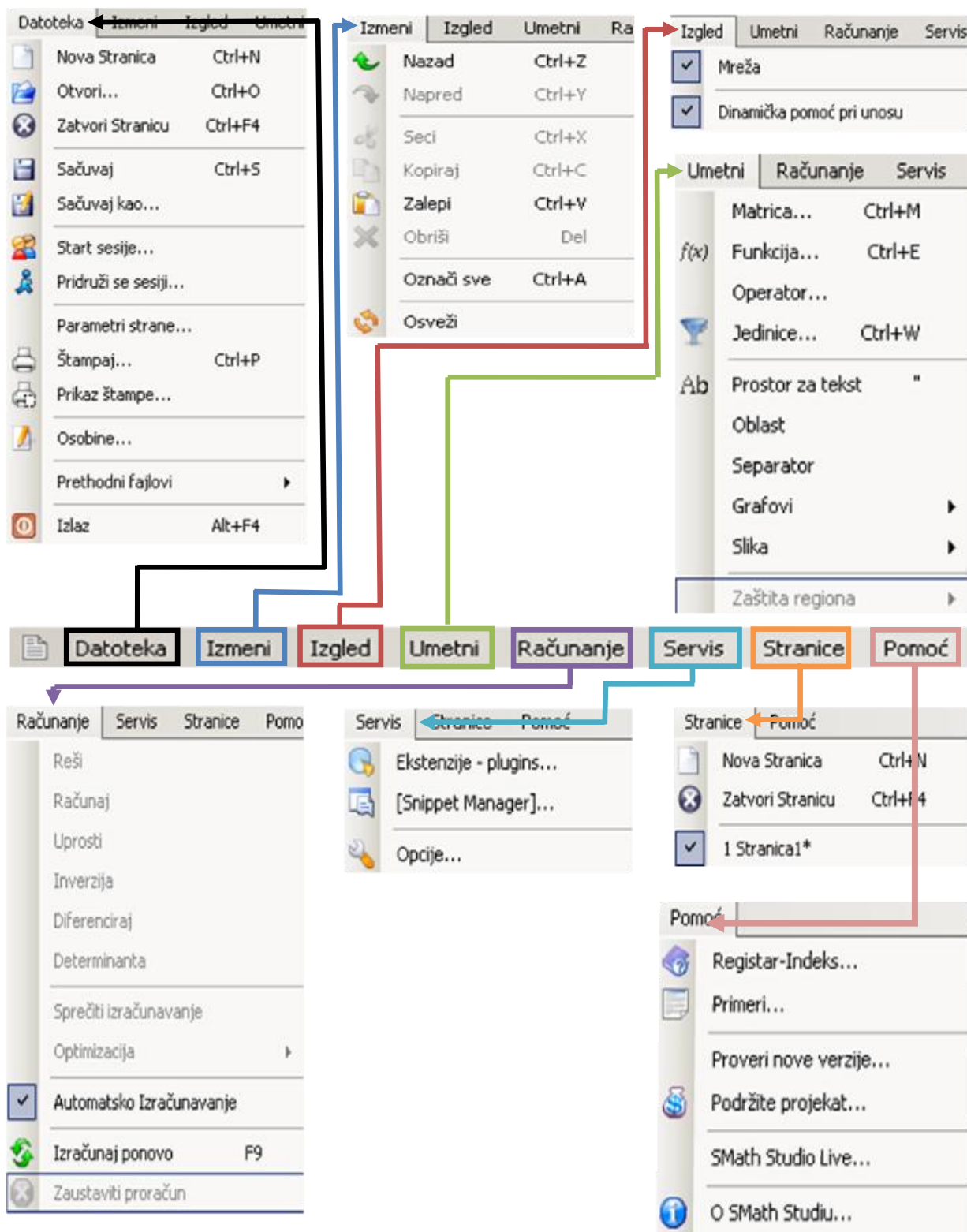
Slika 3. Promena jezika programa 2/2

Sada možemo preći na pregled radnog ekrana.

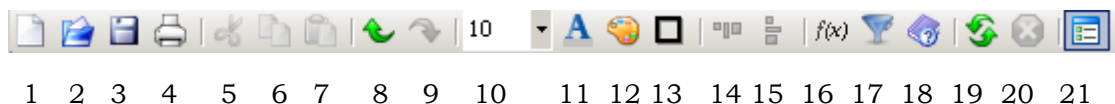
Padajući meniji su sledeći :

- Datoteka
- Izmeni
- Izgled
- Umetni
- Računanje
- Servis
- Stranice
- Pomoć.

Na sledećoj strani prikazani su svi padajući meniji, dok će samo objašnjenje komandi biti prikazano naknadno.



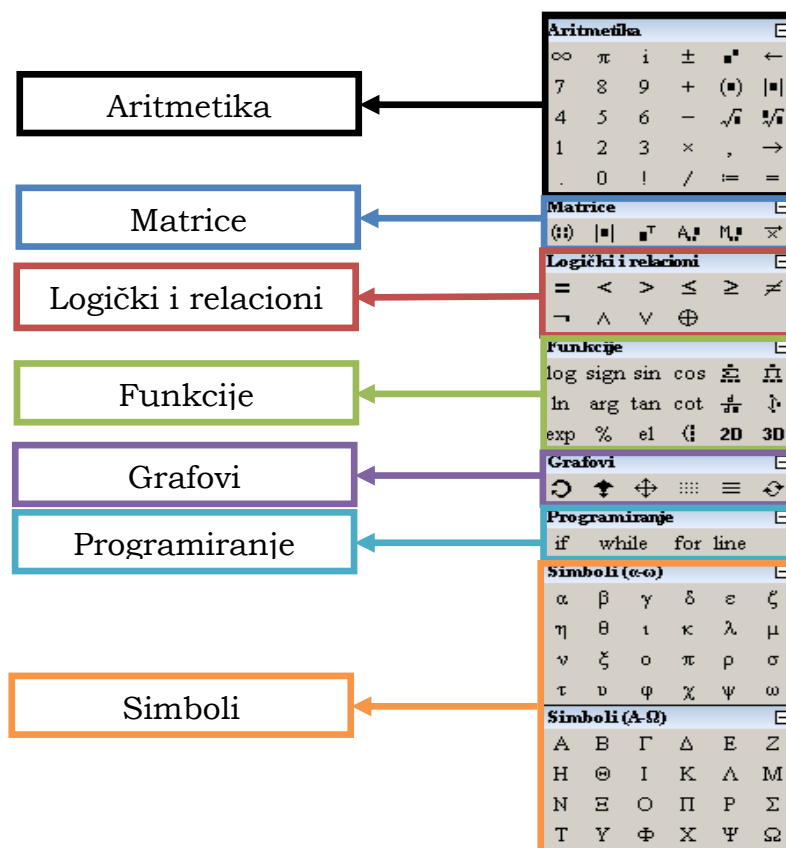
Slika 4. Prikaz padajućih menija



Slika 5. Prikaz toolbar-a

Redom će biti nabrojane komande na toolbar-u :

1. Nova stranica
2. Otvori
3. Sačuvaj
4. Štampaj
5. Seci
6. Kopiraj
7. Zalepi
8. Nazad
9. Napred
10. Veličina teksta
11. Boja teksta
12. Boja pozadine
13. Granica regiona
14. Poravnaj horizontalno
15. Poravnaj vertikalno
16. Funkcija
17. Jedinici
18. Registar – indeks
19. Izračunaj ponovo
20. Zaustavi proračun
21. Prikaži/skloni paletu sa operatorima (paleta sa desne strane).



Slika 7. Prikaz paleta

Regioni

Pošto smo se upoznali sa izgledom programa krenimo sa daljim objašnjenjima. Na sam radni list možemo unositi različite regione :

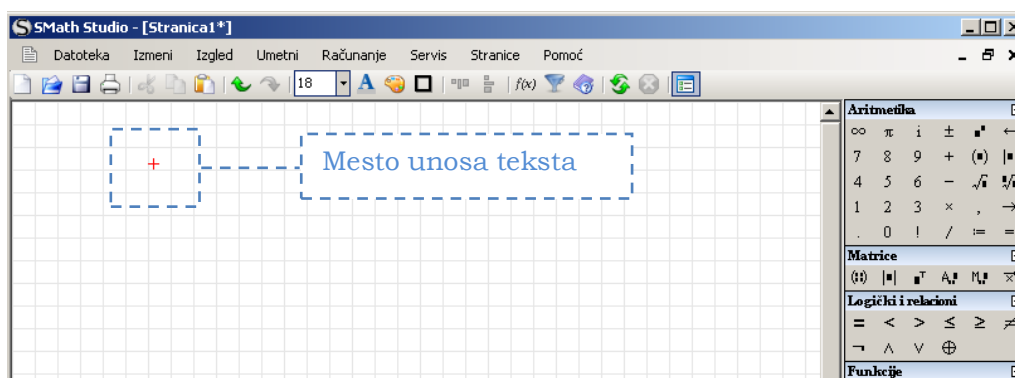
1. Tekstualni – unos komentara, naslovi
2. Matematički – proračun
3. Slike – unos slika u radni list
4. Grafike – prikaz.

SMath se razlikuje od ostalih programa za matematičke proračune u prikazu i unosu podataka. Naime unos u sam program je kao da “pišete” u svesci.

Tekstualni region

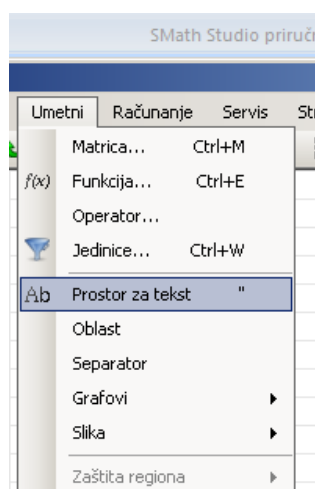
Ako bismo želeli da unesemo tekst na radni list to bi mogli da uradimo na više načina.

Prvo kliknemo na radni list gde bi smo želeli da unesemo tekst. Na radnom listu će se pojaviti crveni krstić koji nam pokazuje gde će se tekst nalaziti.



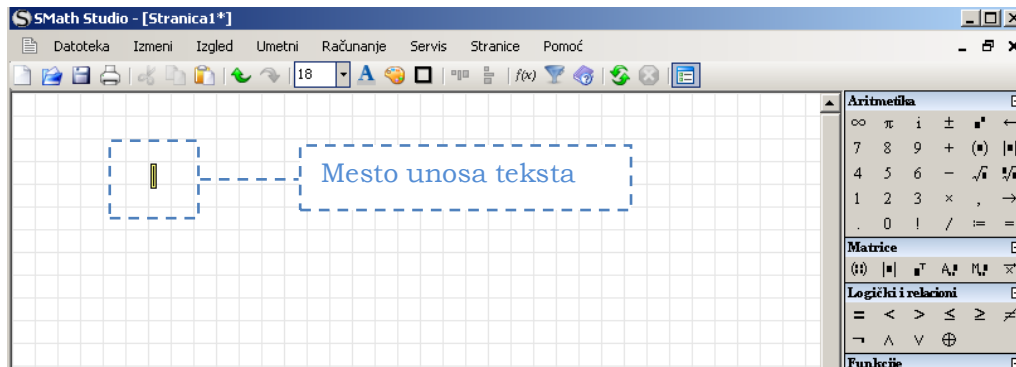
Slika 8. Unošenje tekstualnog regiona 1

Zatim iz padajućeg menija **Umetni** izabrati komandu **Prostor za tekst**.



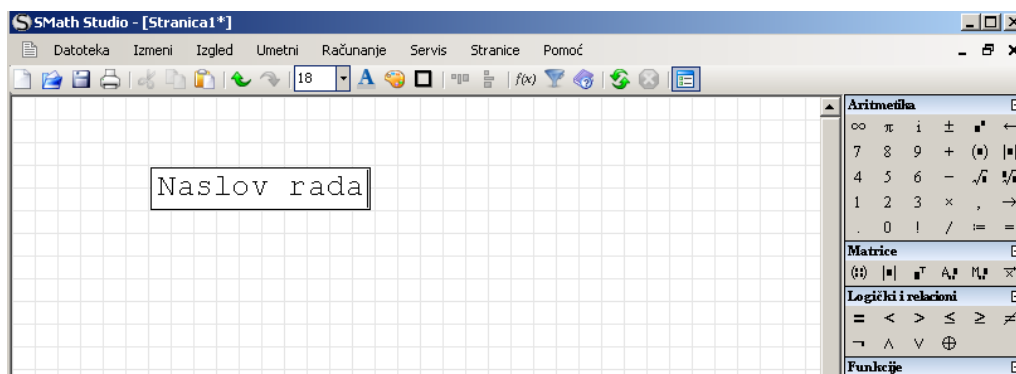
Slika 9. Unošenje tekstualnog regiona 2

Na radnom listu se pojavljuje sledeći prikaz.



Slika 10. Unošenje tekstualnog regiona 3

Zatim počnemo da kucamo željeni tekst.

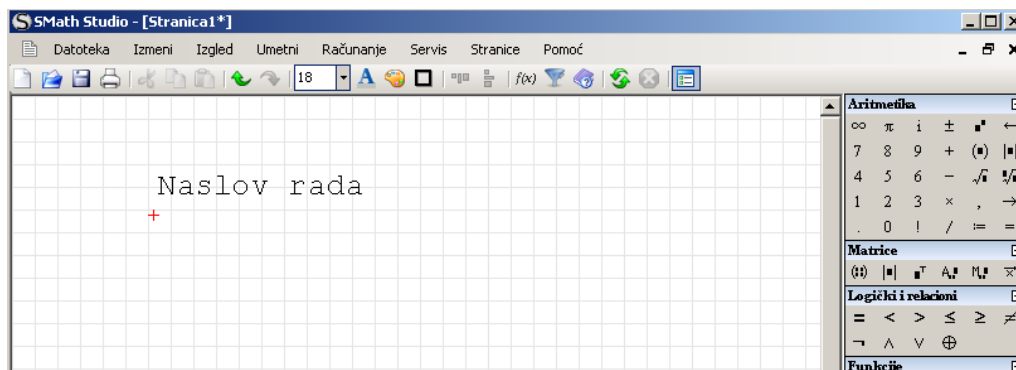


Slika 11. Unošenje tekstualnog regiona 4

Drugi način je da kliknemo na radni list i sa tastature kucamo znak navoda i time bi automatski došli do Slika 10. Unošenje tekstualnog regiona 3.

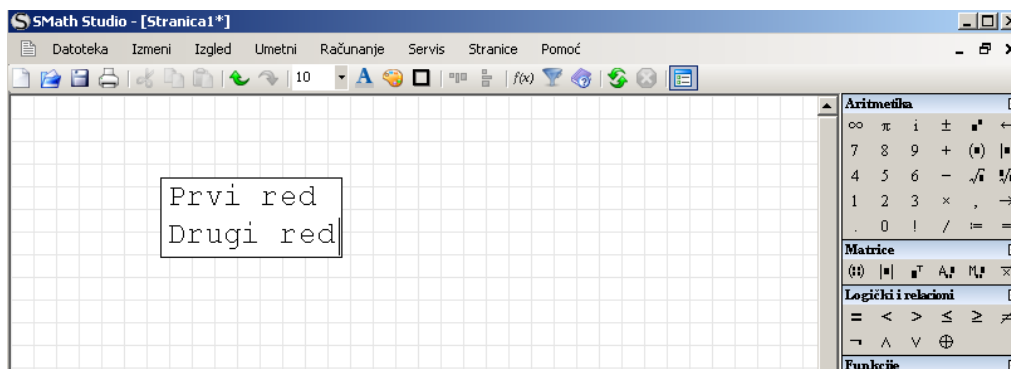


Pritiskom na taster enter izlazimo iz regiona.



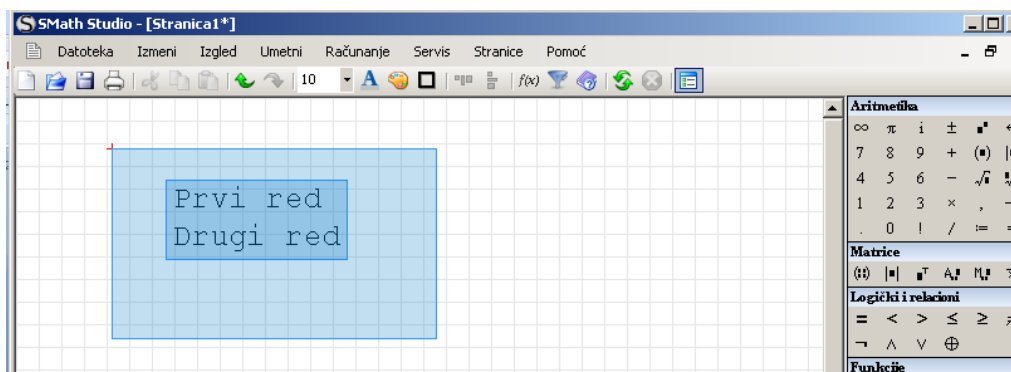
Slika 12. Unošenje tekstualnog regiona 4

Tekst može da se napiše i u nekoliko redova tako što ćemo kada želimo da pređemo u novi red pritisnemo kombinaciju tipki **Ctrl+Enter**.



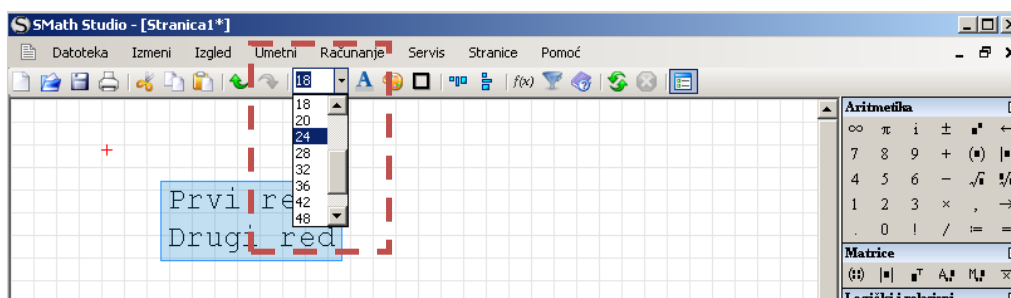
Slika 13. Unošenje tekstualnog regiona 5

Sada možemo krenuti na formatiranje teksta. Prvo moramo da selektujemo tekstualni region.



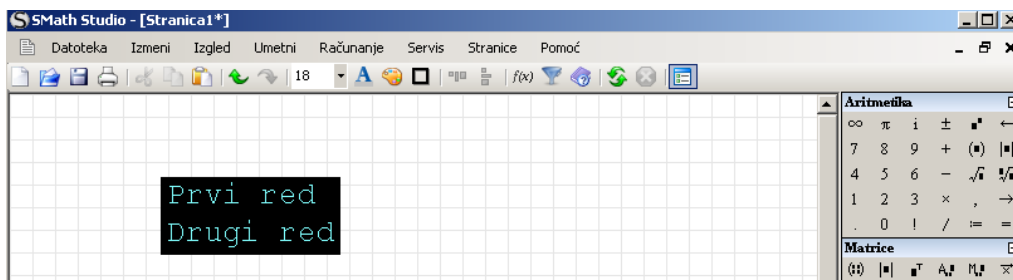
Slika 14. Formatiranje teksta

Možemo uraditi sledeća formatiranja. Povećanje veličine teksta izborom veličine primenom padajućeg menija **Veličina teksta**.



Slika 15. Formatiranje teksta 2

Možemo primenom komada **Boja teksta** i **Boja pozadine** promeniti boju teksta i pozadine.

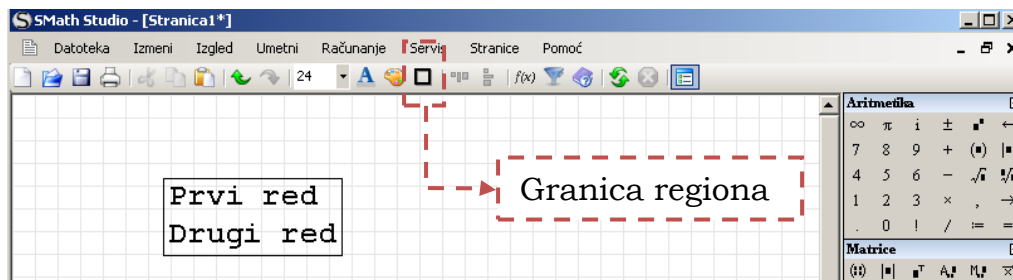


Slika 16. Formatiranje teksta 3

Na raspolaganju su nam i sledeće komande kada je tekst selektovan.

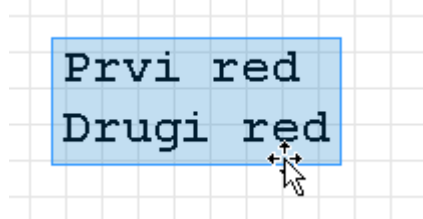
- Ctrl + B** - podebljan tekst (Bold)
- Ctrl + U** - podvučen tekst (Underline)
- Ctrl + I** - iskošen tekst (Italic).

Možemo uneti da su granice regiona vidljive pritiskom na komadu **Granica regiona**.



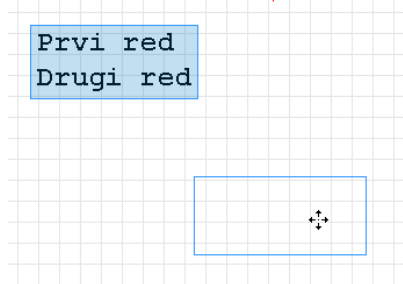
Slika 17. Formatiranje teksta 4

Tekst se može pomerati po stranici tako što ćemo ga prvo selektovati. Pustićemo taster miša, a zatim isti približiti tekstu dok se ne pojavi crni krstić.



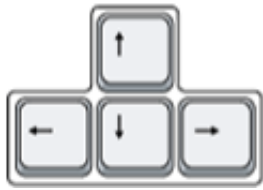
Slika 18. Formatiranje teksta 5

Zatim pritisnemo levi taster miša i pomerimo tekst na željeno mesto.



Slika 19. Formatiranje teksta 6

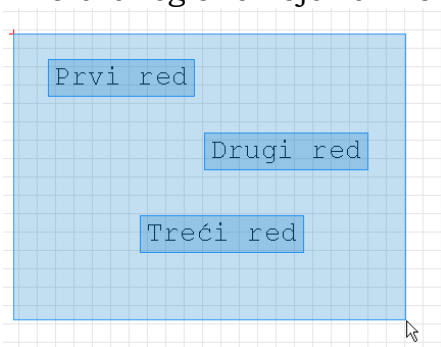
Drugi način je da obeležimo tekst, a samo pomeranje izvršimo korišćenjem strelica sa tastature.



Slika 20. Formatiranje teksta 7

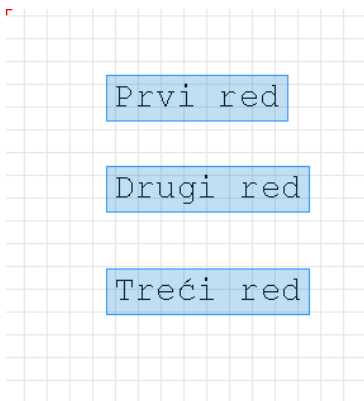
Sledeća slika prikazuje korišćenje komande **Složi regione vertikalno** i **Složi regione horizontalno**.

Prvo sve regione koje želimo da formatiramo selektujemo.



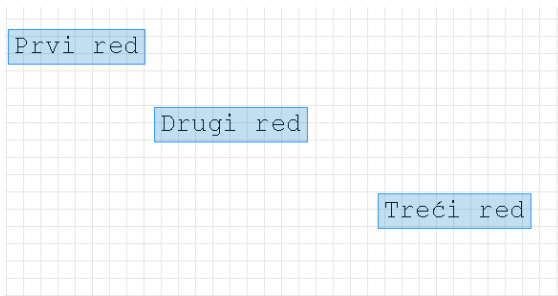
Slika 21. Formatiranje teksta 8

A zatim pritisnemo ikonu **Složi regione vertikalno** . Dobija se sledeće.




Slika 22. Formatiranje teksta 9

Na sličan način radi i komanda Složi regione horizontalno.



Slika 23. Formatiranje teksta 11

Posle upotrebe komande  dobijamo sledeće.



Slika 24. Foramatiranje teksta 12

Naravno na raspolaganju su nam i sledeće komande :

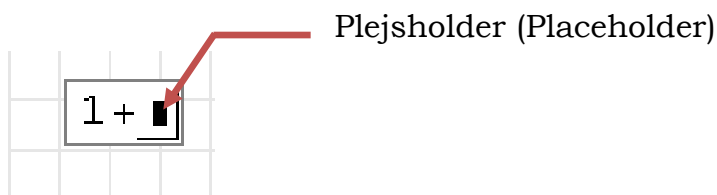
	Seci	Ctrl+X
	Kopiraj	Ctrl+C
	Zalepi	Ctrl+V
	Obriši	Del

Sve što je ranije rečeno o formatiranju tekstualnog regiona važi i za ostale vrste regiona (pomeranje,brisanje,formatiranje).

Unos matematičkih vrednosti

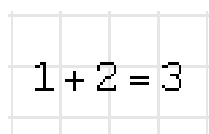
Verujem da ste imali prilike da do sada koristite program Microsoft Word i da ste imali potrebu da unosite određene matematičke izraze. Tom prilikom ste verovatno koristili program Microsoft Equation ili ako koristite Word 2007 ili noviji samo Equation. Unos vrednosti u SMath je veoma sličan unosu u ovim programima. Naravno ovo je sve pod pretpostavkom da koristite Microsoft Windows operativni sistem.

Ako npr. Unesemo izraz $1+2=$ nakon znaka jednakosti SMath će nam prikazati vrednost ovog izraza. Malo analizirajmo ovaj unos. Možemo primetiti da nakon unosa $1+$ pojavljuje se jedan crni pravougaonik. Ovaj pravougaonik se naziva plejsholder i on označava da program očekuje da unesete neku vrednost.



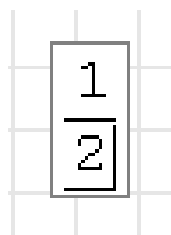
Slika 25. Matematika 1

Nakon unosa broja 2 i znaka jednakosti dobija se:



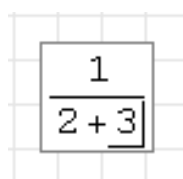
Slika 26. Matematika 2

Pogledajmo još jedan primer. Unesimo vrednost $1 / 2$



Slika 27. Matematika 3

Ako sada unesemo $+3$, dobićemo sledeći izraz.



Slika 28. Matematika 4

Zašto se ovo desilo? Ili je možda bolje pitanje kako se desilo. Odgovor leži u liniji za editovanje, naime na slici 27. možemo primetiti da se ova linija nalazi iza broja 2 što nam govori gde će se unositi izraz ako nastavimo da kucamo.

Ako bismo hteli da izraz sa slike 28. napišemo drugačije to možemo učiniti na sledeći način. Nakon unosa broja 2 pritisnemo stelicu za levo (ili space) i dobićemo sledeći izraz.

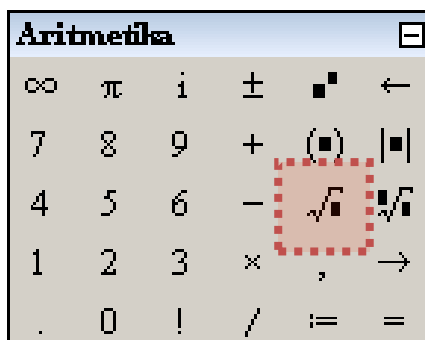
Slika 29. Matematika 5

Sada vidimo da je linija za editovanje obuhvatila ceo izraz tj. vidimo gde će se unosti vrednosti. Ako sada ukucamo +3 dobijamo:

Slika 30. Matematika 6

Ako želimo da unesemo neku aritmetičku operaciju to možemo učiniti ili preko palete sa leve strane ili preko tastature.

Primer: Ako želimo da izračunamo kvadratni koren od broja 6 to možemo učiniti unosom sa palete **Aritmetika**.

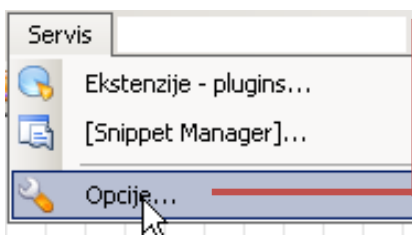


Slika 31. Paleta aritmetika

Ili preko tastature kucajući $\sqrt{6} =$. Dobija se sledeće:

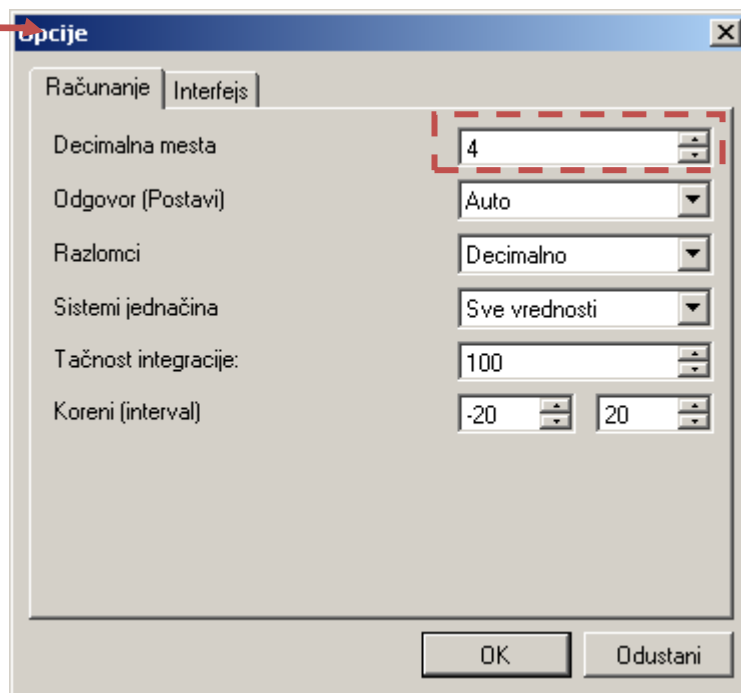
Slika 32. Matematika 7

Promena broja decimalnih mesta se može izvršiti izborom komande **Opcije** sa padajućeg menija **Servis**.

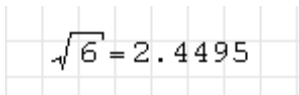


Slika 33. Promena broja decimala

Povećali smo broj decimala sa dve na četiri.



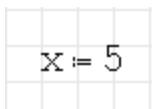
Sada je prikaz sledeći.



Slika 34. Matematika 8

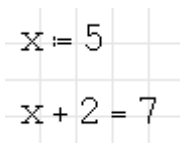
Rad sa varijablama (promenljive)

Ako otkucamo $x:5$ na radnom listu će biti prikazano.



Slika 35. Promenljiva 1

Ovo znači da smo promenljivoj x pridružili vrednost 5. Ako bi iskucali $x+2=$ kao rezultat bi dobili:



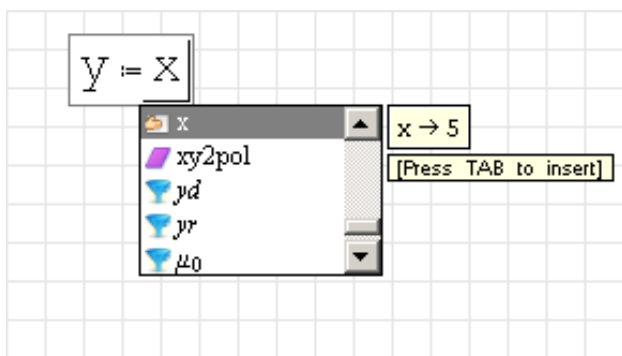
Slika 36. Promenljiva 2

Ovo znači da možemo menjati vrednost promenljive x tako što ćemo obrisati vrednost 5 i ukucati neku drugu, a program će automatski izračunati donji izraz.

Interesantno, ali čemu to? Ako imamo neki komplikovaniji izraz i naravno više promenljivih lako možemo menjajući neku od njih dobijati rezultate bez potrebe da pišemo ponovo ceo proračun.

Hajde sada da definišemo novu promeljivu koja zavisi od pomenljive x. Neka to bude $y:x+3.5$.

Prilikom unosa ovog izraza možemo primetiti da nakon pritiska tastera x pojavljuje se dinamički asistent koji nam pomaže pri unosu.



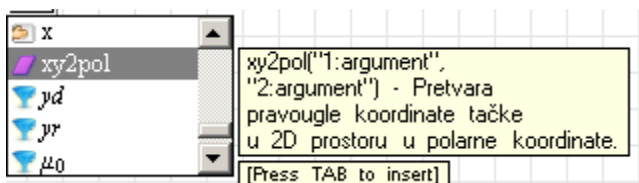
Slika 37. Dinamički asistent 1

On nam prikazuje nekoliko informacija i to da postoji promenljiva x koja je ranije definisana i da je njena vrednost 5.

Dinamički asistent nam pomaže prilikom različitih unosa tako što nam :

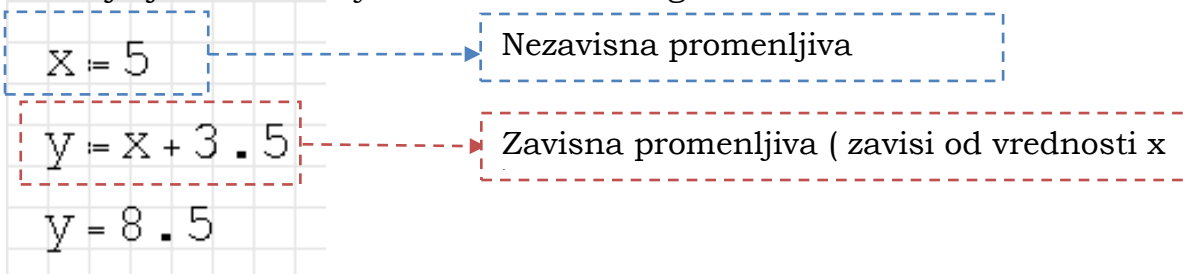
- Prikazuje izraz koje je definisao sam korisnik
- Ugrađene jedinice (kN, m)
- Programski definisane konstante
- Funkcije

On nam služi kao neka vrsta prečice. Usput dobijamo informacije za šta služi izabrana opcija.



Slika 38. Dinamički asistent 2

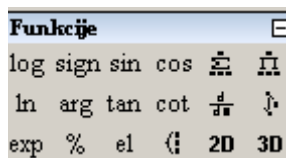
Na kraju još unesemo $y=$. Ceo unos sad izgleda ovako.



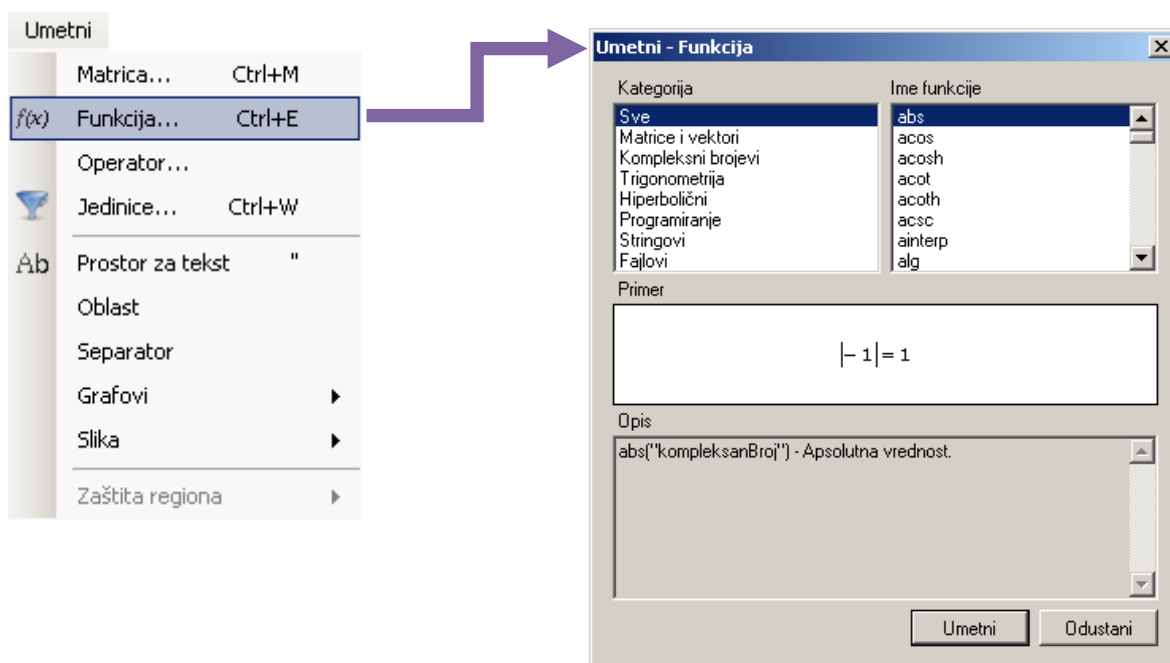
Slika 39. Matematika 9

Funkcije

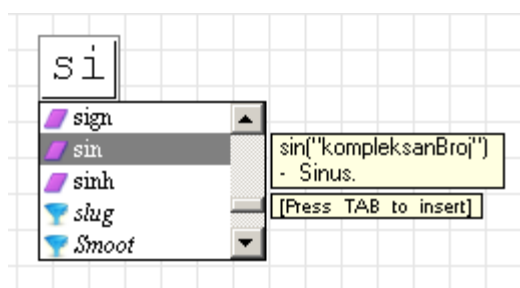
Funkcije u programu SMath Studio možemo unositi na nekoliko načina. Odabirom na paleti **Funkcije**, preko padajućeg menija **Umetni/Funkcija** ili kucanjem same funkcije (naravno ako joj znamo naziv).



Slika 40. Funkcije paleta 1

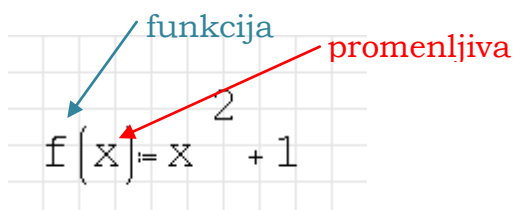


Slika 41. Funkcije 2



Slika 42. Funkcije 3

Možemo definisati i svoje funkcije. Npr. funkcija $f(x) := x^2 + 1$



Slika 43. Funkcija 4

Sada možemo pozivati funkciju $f(x)$ za različite vrednosti promenljive x .

$$f(x) = x^2 + 1$$

$$f(5) = 26$$

$$f(2) = 5$$

$$f(-150) = 22501$$

Slika 44. Funkcija 5

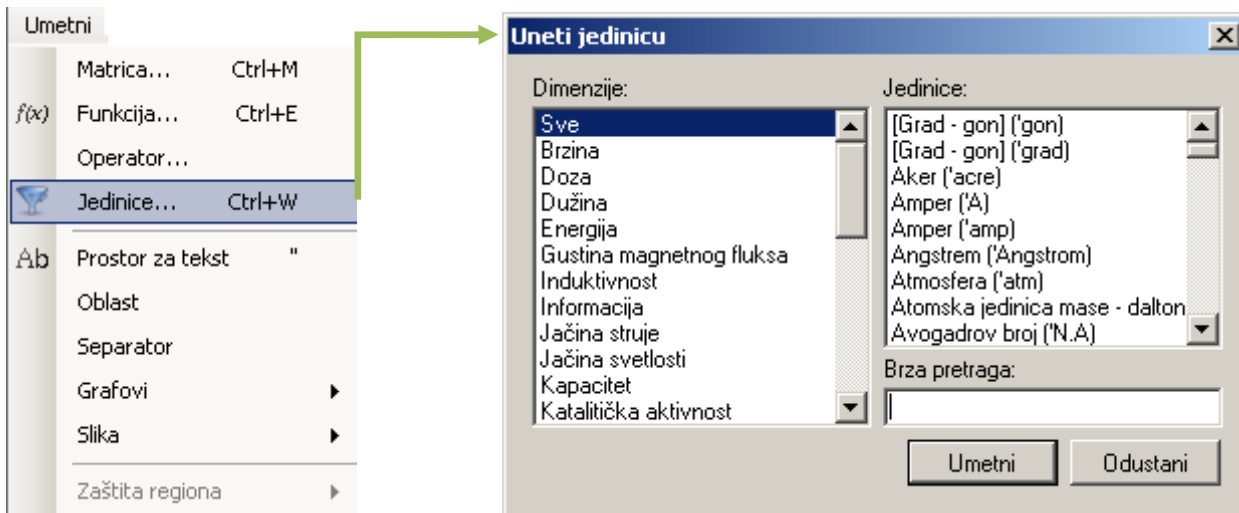
Definisanje jedinica

SMath Studio ima ugrađeno veliki broj jedinica. Unos jedinice se može uraditi na različite načine. Ako bismo na primer hteli da unesemo da je neka promenljiva $P1$ jednaka 100 N to bi mogli uraditi na sledeći način. Ukucali bi $P1:100$ i program bi prikazao sledeće:

$$P1 := 100$$

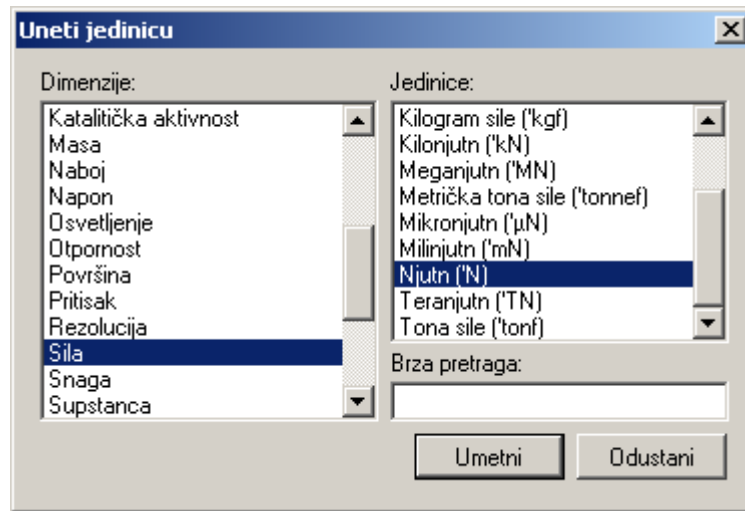
Slika 45. Jedinice 1

Sada dolazi na red unos jedinice N. Iz padajućeg menija **Umetni** izaberemo komandu **Jedinici**. Pojaviće se sledeći prozor:



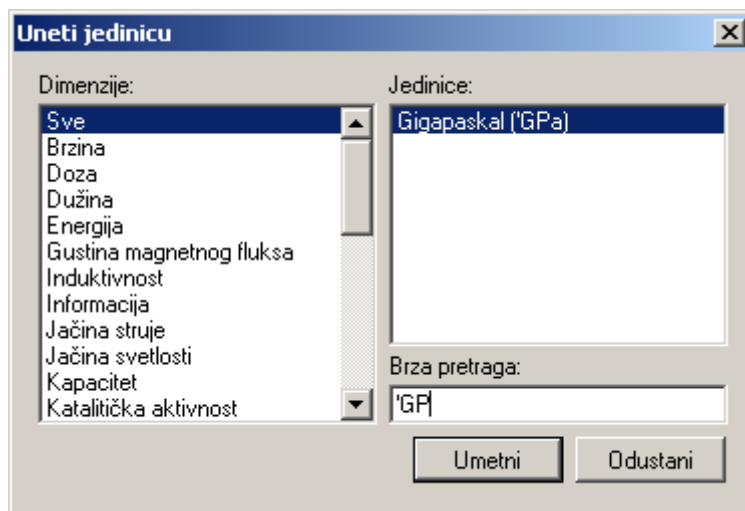
Slika 46. Jedinice 2

Na desnom delu ovog prozora nalaze se dimenzije tj. grupa , a sa leve strane same jedinice. Izborom grupe **Sila** i jedinice **Njutn** , a zatim pritiskom na dugme **Umetni** izvršili smo unos jedinice.



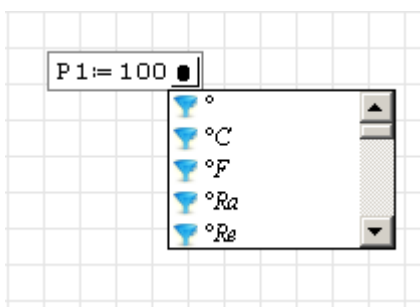
Slika 47. Jedinice 3

U ovom prozoru postoji i opcija brza pretraga u kojoj se mogu kucati nazivi jedinica ili kucati jednostruki navodnik (') i oznaku jedinice i tako je brzo pronaći.



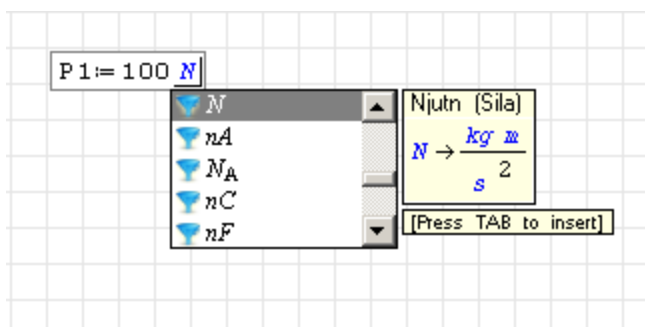
Slika 48. Jedinice 4

Drugi i lakši način unošenja jedinice je da nakon ukucanih 100 pritisnemo znak jednostrukog navoda ('). Na ekranu će se pojaviti sledeće:



Slika 49. Jedinice 5

Iza broja 100 pojavio se zaobljen pravougaonik i on predstavlja plejsholder za jedinice (unite placeholder). Sada samo ukucamo slovo N.



Slika 50. Jedinice 6

Primer 1 :

```
P1:= 100 N
P2:= 50 N
P1+P2= 150 N
```

Ukucali smo dve promenljive P1 i P2, definisali ih a zatim sabrali. Rezultat je iste jedinice kao i promenljive. Ako bismo želeli da rezultat predstavimo u drugim jedinicama to bismo uradili na sledeći način.

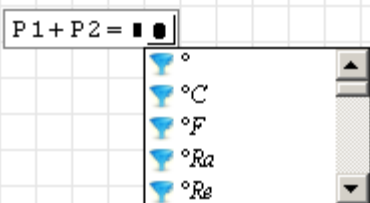
Postavili bi smo liniju unosa posle jedinice rezultata.

```
P1+P2= 150 N
```

Tada bi se posle jedinice N pojavio još jedan plejsholder. Ako liniju pomerimo do njega pomoću strelica :

```
P1+P2= 150 N
```

A zatim otkucamo jednostruki navod ('):



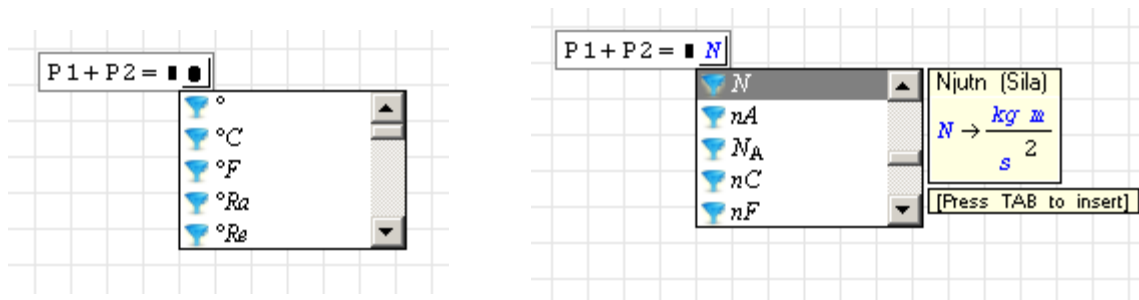
Sada samo unesemo željenu jedinicu , u ovom slučaju to je kN.

```
P1+P2=0.15 kN
```

Ako bi sad želeli da vratimo sa kN ponovo na N. Ponavljajući korake koji su gore prikazani uočili bismo sledeće. Kada postavimo liniju unosa iza jedinice kN ne pojavljuje se plejsholder za unos jedinica.

$$P1+P2=0.15 \text{ kN}$$

Menjanje jedinica u ovom slučaju se svodi na to da obrišemo jedinicu kN i potom ukucamo nove jedinice.



Primer 2 :

Ukucajmo sledeće $10\text{'m}+5\text{'ft} =$

$$10 \text{ m} + 5 \text{ ft} = 11.52 \text{ m}$$

Iz ovog primera zaključujemo da možemo sabirati različite jedinice u ovom slučaju za dužinu.

$$10 \text{ m} + 5 \text{ ft} = 453.7 \text{ in}$$

Menjajući jedinicu rezultata iz metar u inče vidimo da će SMath Studio sam izvršiti konverziju jedinica.

Program dozvoljava mogućnost da sami definišemo svoje jedinice. Kao ilustracija poslužićemo se primerom definisanja jedinice xN koja će biti jednaka 100 kN. Počecemo da kucamo sledeći izraz 'xN:100'kN

$$xN := 100 \text{ kN}$$

Ovim postupkom smo definisali novu jedinicu xN. Pogledajmo par primera sa ovom jedinicom.

$$xN := 100 \text{ kN}$$

$$50 \text{ xN} + 20 \text{ kN} = 5.02 \cdot 10^6 \text{ N}$$

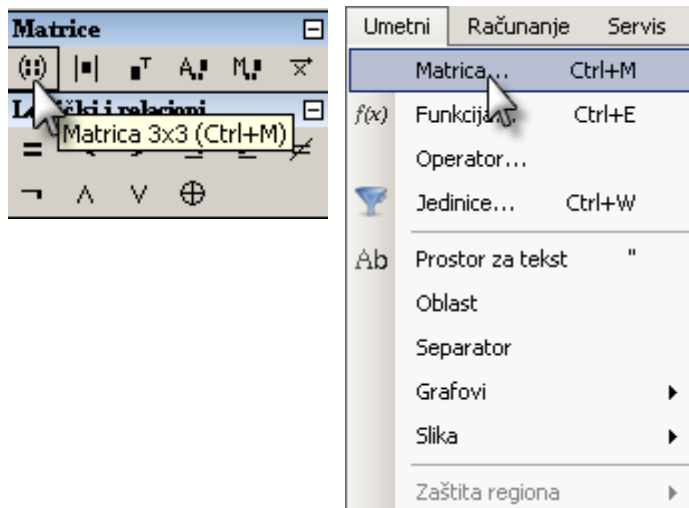
$$50 \text{ xN} + 20 \text{ kN} = 50.2 \text{ xN}$$

$$50 \text{ xN} + 20 \text{ xN} = 7 \cdot 10^6 \text{ N}$$

$$50 \text{ xN} + 20 \text{ xN} = 70 \text{ xN}$$

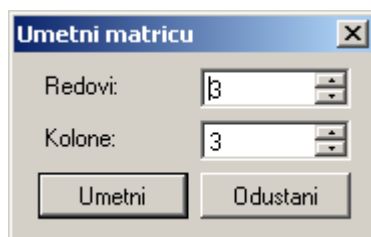
Matrice i vektori

Unos matrice može se postići ili preko panela ili preko padajućeg menija **Umetni/Matrica**.



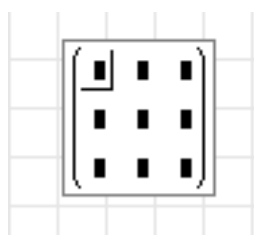
Slika 51. Matrice 1

Program će zatim zahtevati da se unese broj redova i kolona buduće matrice.



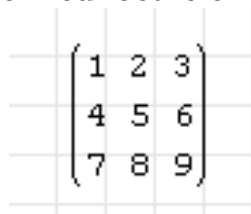
Slika 52. Matrice 2

Unošenje matrice postizemo pritiskom na dugme **Umetni** na radnom listu se pojavljuje sledeći prikaz.



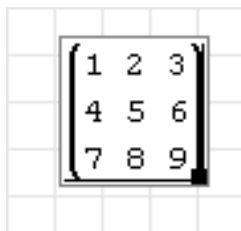
Slika 53. Matrice 3

U prazna polja ukucavamo brojne vrednosti elemenata matrice.



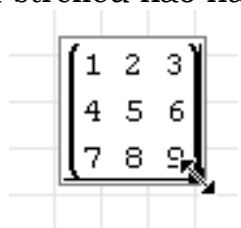
Slika 54. Matrice 4

Ako bismo želeli da proširimo ovu matricu tj. da joj dodamo (oduzmemo) red i/ili kolonu to možemo učiniti na sledeći način. Dovedemo liniju unosa pomoću strelica ili protiskom na taster **Space** da obuhvati celu matricu.



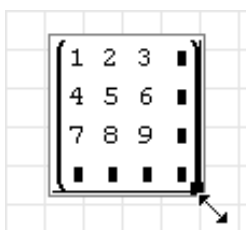
Slika 55. Matrice 5

Vidimo da se u desnom donjem uglu pojavljuje crna kockica. Kada se približimo ovoj kockici kursor menja oblik u strelicu kao na slici 56.



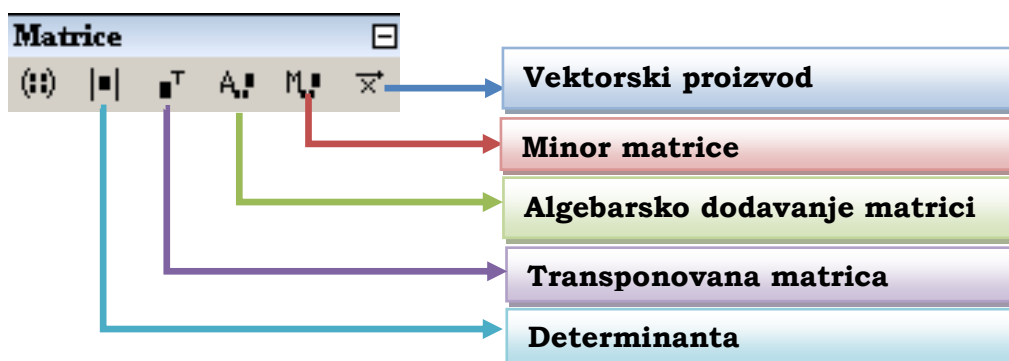
Slika 56. Matrice 6

Pritiskom levog tastera miša možemo proširivati/smanjivati broj redova i kolone. Kada promenimo veličinu matrice puštamo levi taster miša.



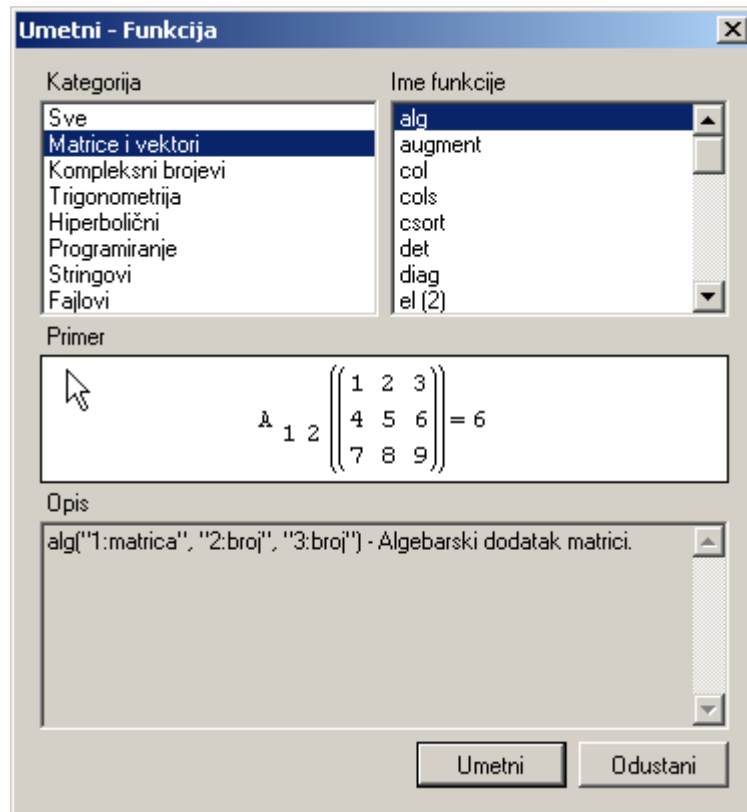
Slika 57. Matrice 7

U programu postoji dosta ugrađenih funkcija koje se mogu primeniti na matricama. Jedan manji broj se nalazi na paleti **Matrice**.



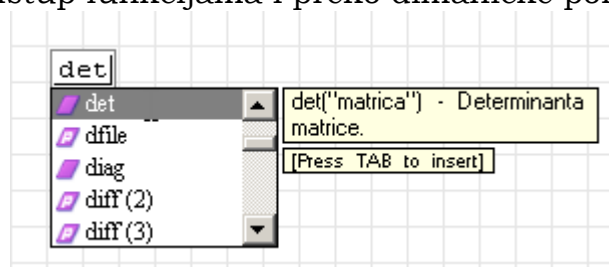
Slika 58. Matrice 8

Sve funkcije možemo naći u padajućem meniju **Umetni/Funkcije** i izborom kategorije **Matrice i vektori**.



Slika 59. Matrice 9

Naravno moguć je pristup funkcijama i preko dimaničke pomoći.



Slika 60. Matrice 10

Primer :

$$A = \begin{pmatrix} 1 & 3 & 9 \\ 12 & 15 & 18 \\ 21 & 24 & 27 \end{pmatrix}$$
 Matrica A

$$|A| = -108$$
 Determinanta matrice

$$A^T = \begin{pmatrix} 1 & 12 & 21 \\ 3 & 15 & 24 \\ 9 & 18 & 27 \end{pmatrix}$$
 Transponovana matrica

$$A^{-1} = \begin{pmatrix} 0.25 & -1.25 & 0.75 \\ -0.5 & 1.5 & -0.83 \\ 0.25 & -0.36 & 0.19 \end{pmatrix}$$
 Inverzna matrica

$$A \cdot A^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Slika 61. Matrica 11

Izdvajanje vrsta i redova matrice postiže se na sledeći način.

$$A = \begin{pmatrix} 1 & 3 & 9 \\ 12 & 15 & 18 \\ 21 & 24 & 27 \end{pmatrix}$$

kolone

redovi

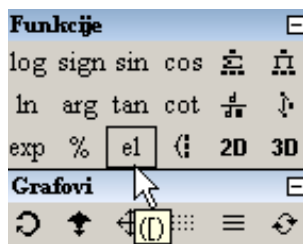
Matrica A

$$\text{col}(A, 1) = \begin{pmatrix} 1 \\ 12 \\ 21 \end{pmatrix}$$
 Izdvoj kolonu br. 1

$$\text{row}(A, 2) = (12 \ 15 \ 18)$$
 Izdvoj red br. 2

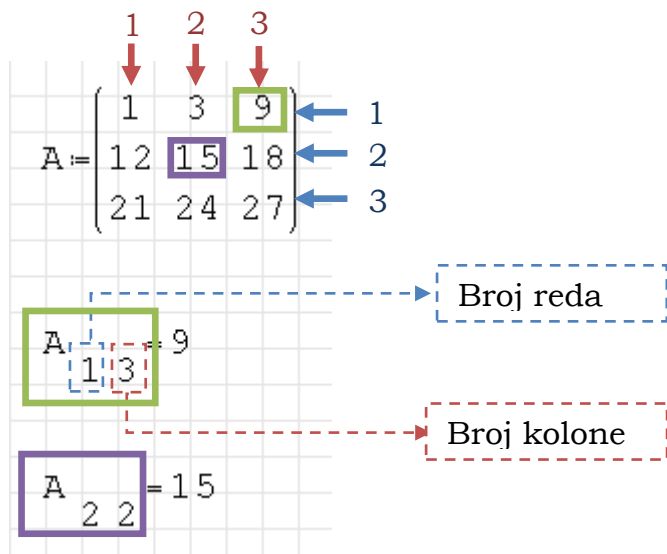
Slika 62. Matrica 12

Izdvajanje elementa matrice vrši se tako što sa palete izaberemo komandu **el**.



Slika 63. Matrica 13

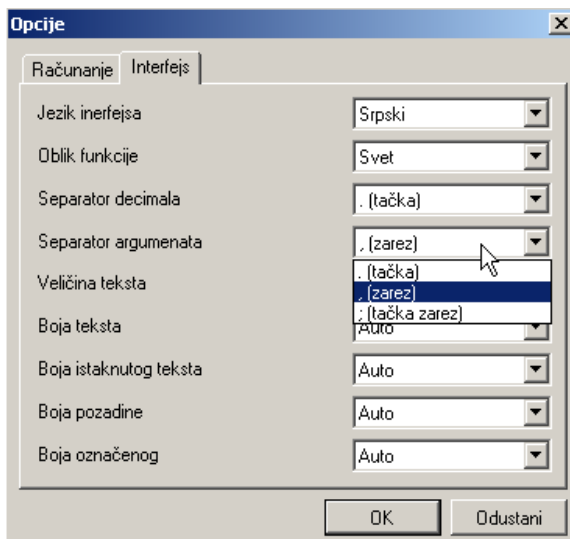
Ili pritiskom na tastaturi tipke **[** . Postupak će biti prikazan na sledećoj slici.



Prvo ukucamo $A[1,3=$ ili umesto znaka **[** sa palete izaberemo **el**

Slika 64. Matrica 14

Separator argumenata u ovom slučaju je zarez ali se ovo može promeniti izborom **Servis/Opcije** i odabirom znaka separatora.



Slika 65. Matrica 15

Rešavanje sistema linearnih jednačina

Primer:

Rešiti sistem linearnih jednačina.

$$2 \cdot x_1 + 3 \cdot x_2 - 2 \cdot x_3 = 15$$

$$3 \cdot x_1 - 2 \cdot x_2 + 2 \cdot x_3 = -2$$

$$4 \cdot x_1 - x_2 + 3 \cdot x_3 = 2$$

$$D := \begin{pmatrix} 2 & 3 & -2 \\ 3 & -2 & 2 \\ 4 & -1 & 3 \end{pmatrix}$$

$$\delta := \begin{pmatrix} 15 \\ -2 \\ 2 \end{pmatrix}$$

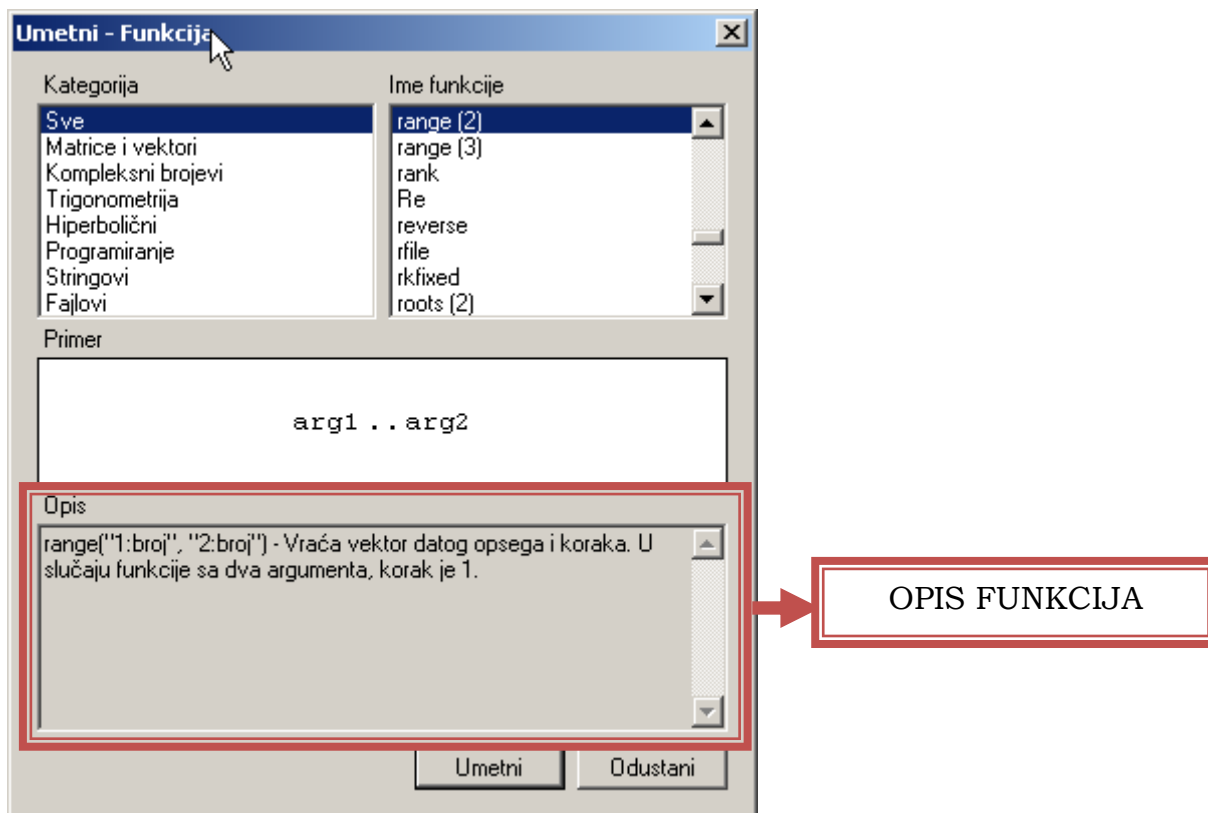
$$X := D^{-1} \cdot \delta$$

$$X = \begin{pmatrix} 2 \\ 3 \\ -1 \end{pmatrix}$$

Slika 66. Matrica 16

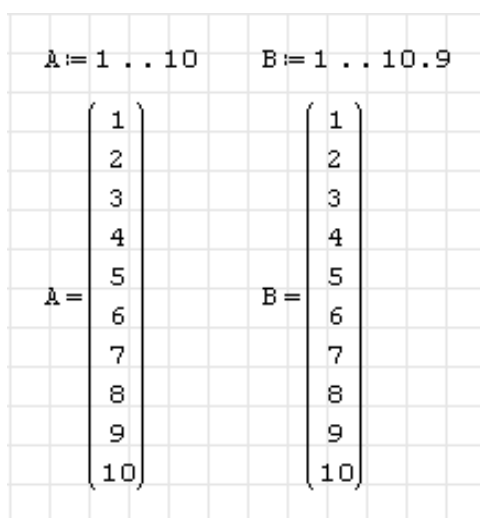
Opseg (Range)

Ova komanda nam vraća vektor vrednosti u zadatom opsegu. SMath poseduje dva oblika ove funkcije **range(2)** i **range(3)**. Opis svake funkcije možemo videti u **Umetni/Funkcija** i zatim pronalaženja ovih funkcija.



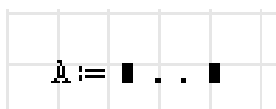
Slika 67. Matrica 17

Na sledećem primeru biće prikazane funkcije **range**.



Kucamo sledeće:

A: (**Umetni funkciju**) **range(2)** ili samo kucamo **range** i iz dinamičkog asistenta biramo **range(2)**. Na ekranu će nam se prikazati sledeće:

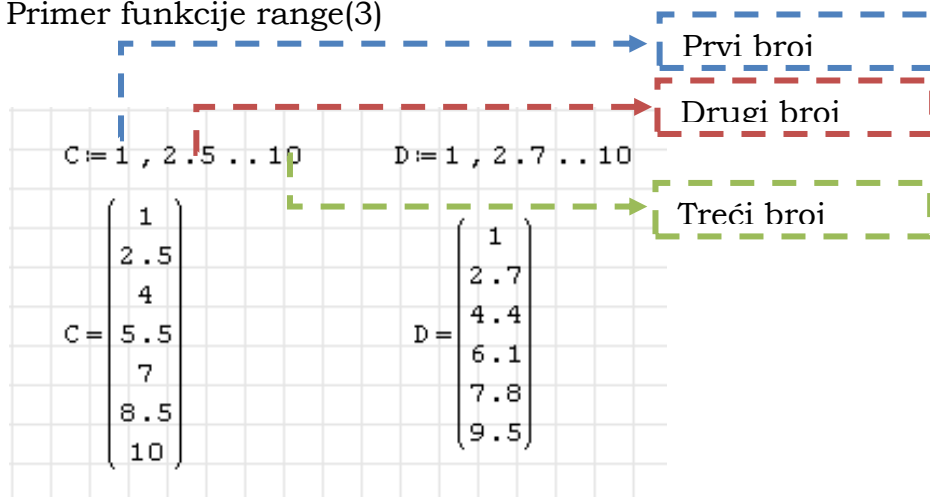


U prazne plejsholdere ukucavamo raspon od 1 do 10.

Slika 68. Opseg 1

U slučaju vektora **B** pošto je inkrement (korak) jednak jedinici poslednji član 10.9 nije prikazan.

Primer funkcije range(3)

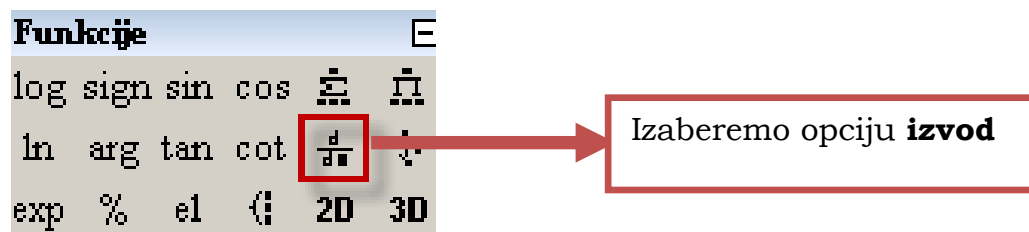


Slika 69. Opseg 2

Kako program računa inkrement u ovom slučaju? Ovo će biti objašnjeno na primeru vektora C. Da bi dobio korak program od drugog broja oduzima prvi. U ovom slučaju $2,5-1=1,5$. Tako da se dobija niz 1; $1+1,5=2,5$; $2,5+1,5=4$; $4+1,5=5,5$; $5,5+1,5=7$; $7+1,5=8,5$; $8,5+1,5=10$. Zadnji broj ne mora da se nađe u vektoru. To prikazuje vektor D gde zadnji član (broj 10) nije dodeljen.

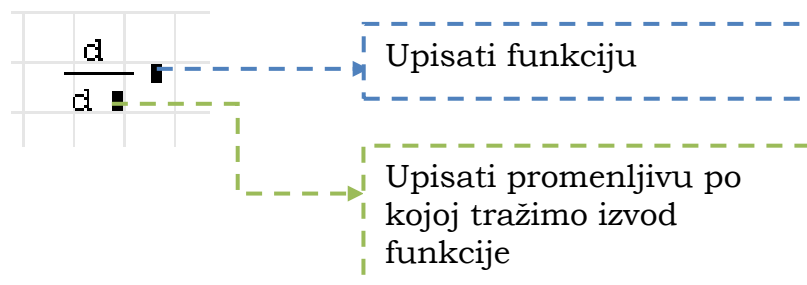
Izvodi i integrali

Ovaj deo ćemo objasniti na par jednostavnih primera.

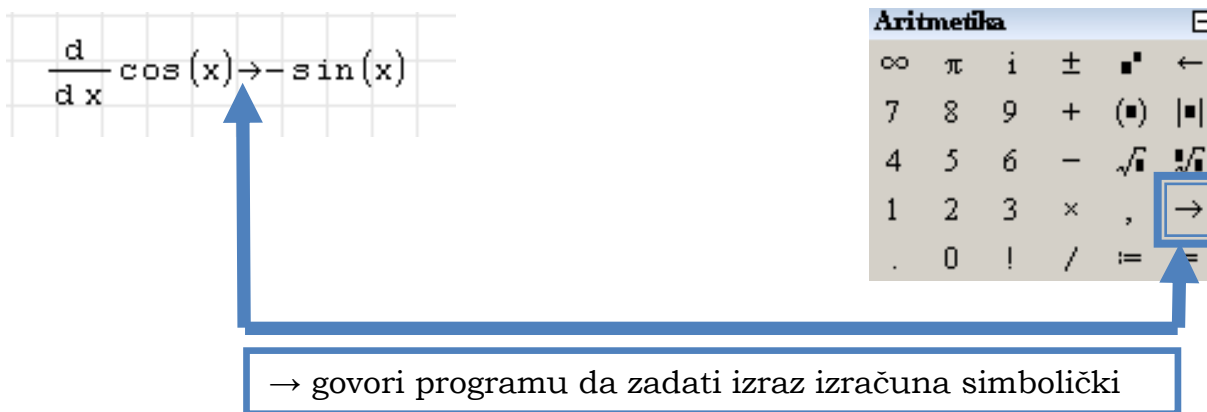


Slika 70. Izvod 1

Na ekranu se pojavljuje sledeće:

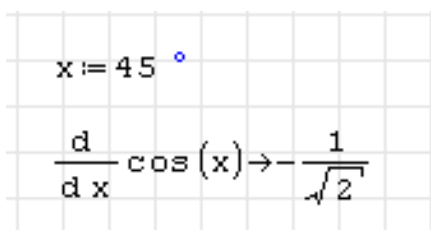


Slika 71. Izvod 2



Slika 72. Izvod 3

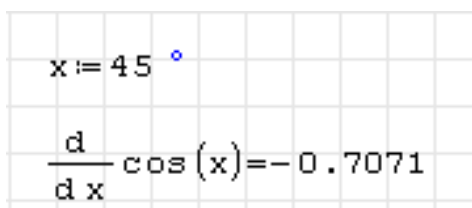
Ako bismo ranije definisali vrednost promenljive x.



Slika 73. Izvod 4

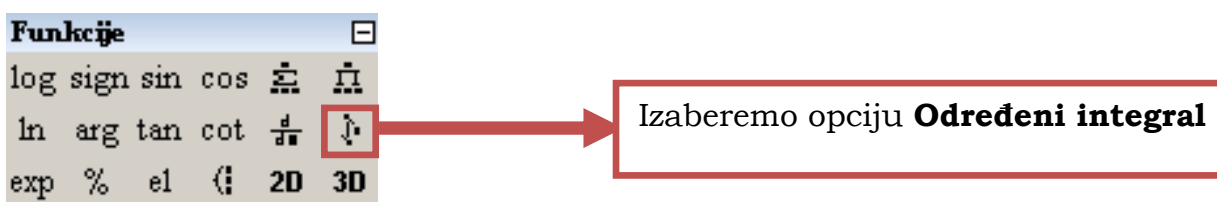
Vidimo da će program prvo izvod rešiti simbolički pa tek onda uvrstiti vrednost za x.

Ovako definisan izvod može da se reši i numerički tako što ćemo umesto \rightarrow staviti znak jednakosti.

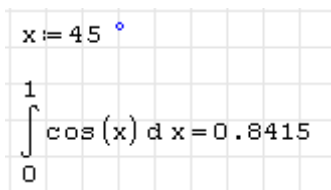


Slika 74. Izvod 5

Za određeni integral sa panela biramo komandu.



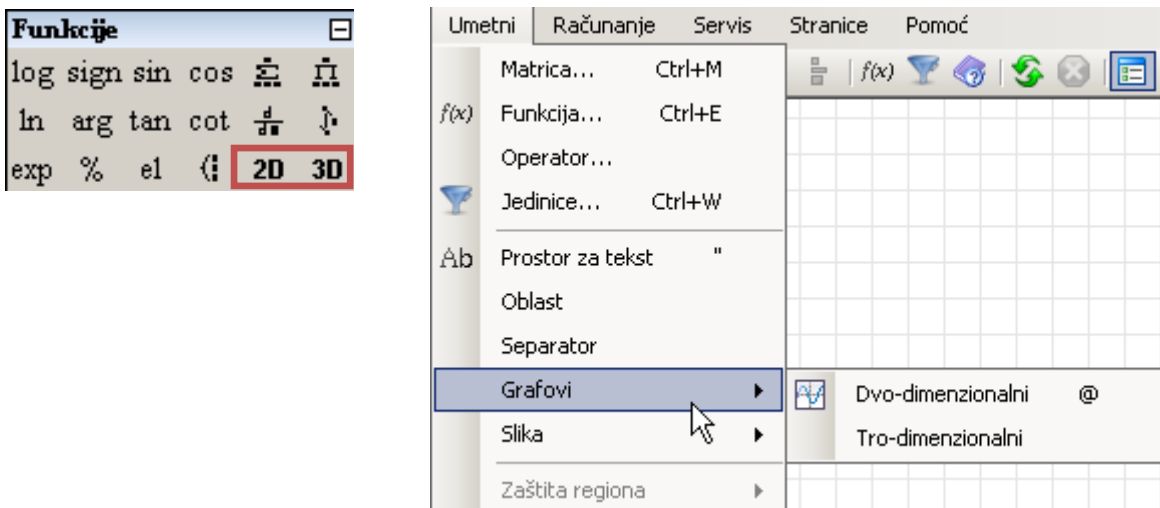
Slika 75. Integral 1



Slika 76. Integral 2

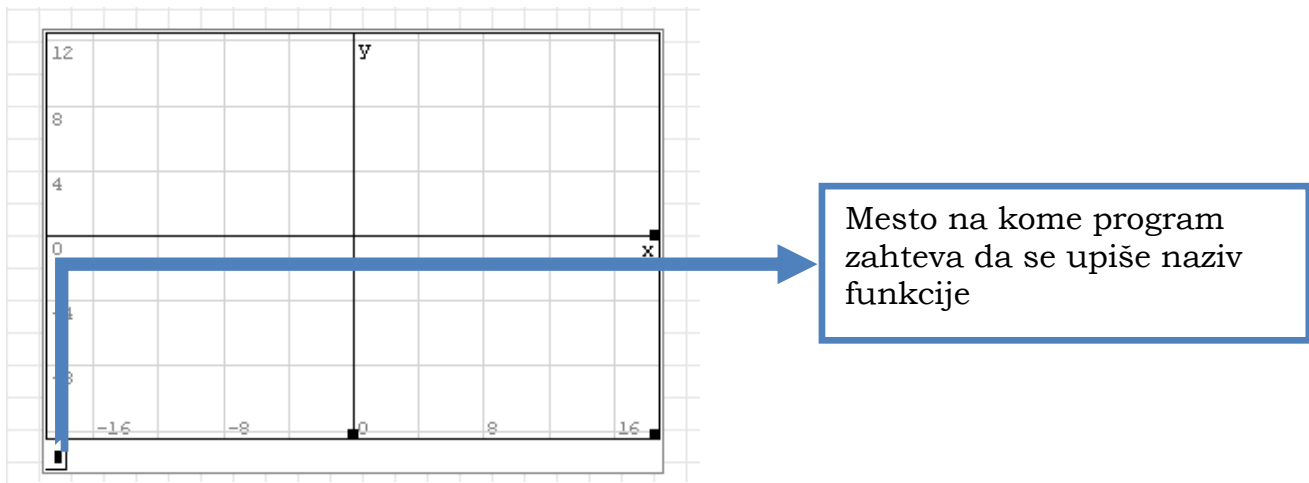
Grafici

SMath studio poseduje mogućnost prikaza 2D i 3D grafika. Grafike možemo unositi ili pritiskom komande **2D** ili **3D** sa palete **Funkcije** ili preko padajućeg menija **Umetni / Grafovi** i zatim izborom vrste grafika.



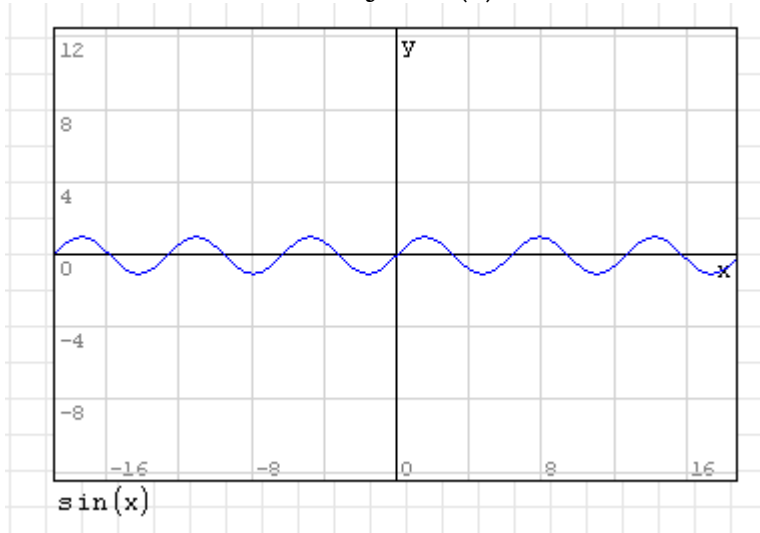
Slika 77. Grafici 1

Prvo ćemo prikazati unos 2D grafika. Pritiskom ili preko panela **Funkcije** ili preko padajućeg menija na radnom listu imamo sledeće.



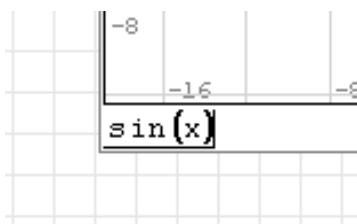
Slika 78. Grafici 2

Mi ćemo izabrati funkciju $\sin(x)$.



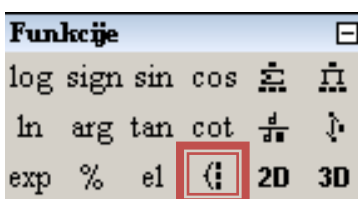
Slika 79. Grafici 3

Kada bismo želeli da na istom grafiku dodamo još jednu funkciju (npr. $\cos(x)$) to bismo uradili na sledeći način. Postavimo liniju unosa (pomoću strelica ili dugmeta space) da označi ceo izraz kao na slici 80.



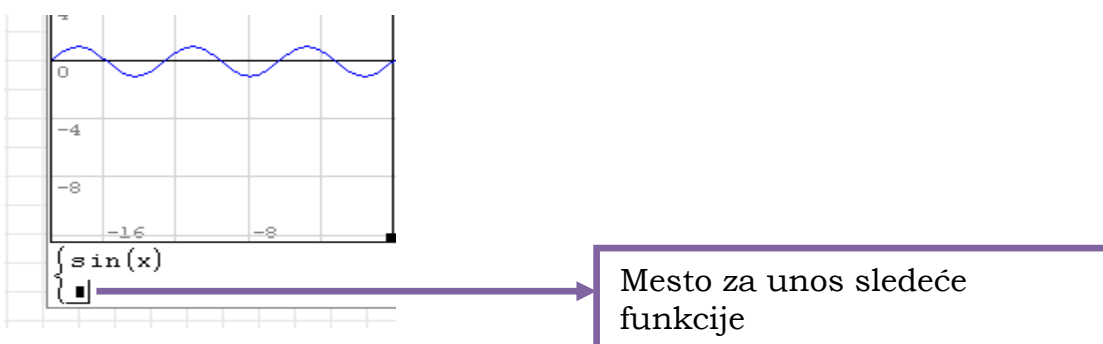
Slika 80. Grafici 4

Zatim sa palete **Funkcije** izaberemo komandu **Sistem jednačina**.

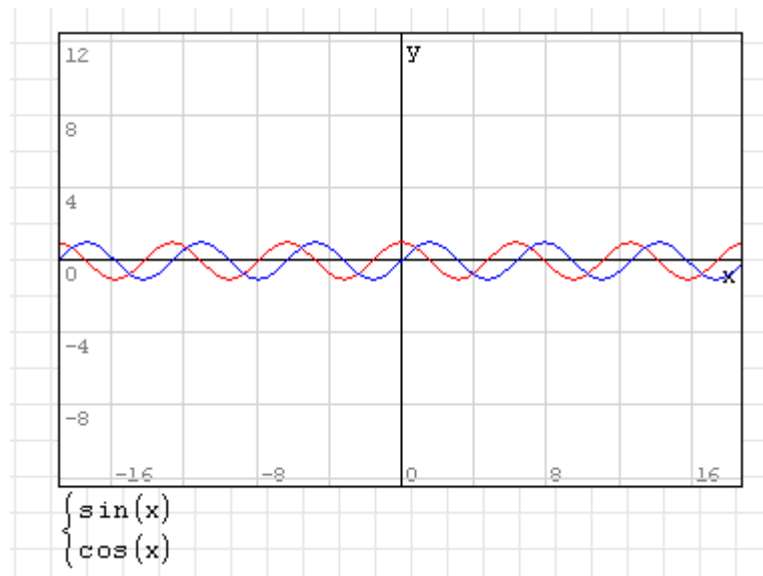


Slika 81. Grafici 5

Ispod grafika će se pojaviti.

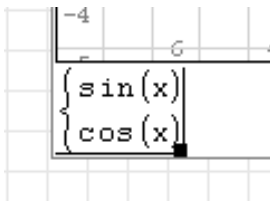


Slika 82. Grafici 6



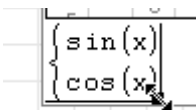
Slika 83. Grafici 7

Za dodavanje još funkcija potrebno je sa linijom unosa obuhvatiti sve funkcije ($\sin(x)$ i $\cos(x)$) dok se ne pokaže crna kockica kao na slici 84.



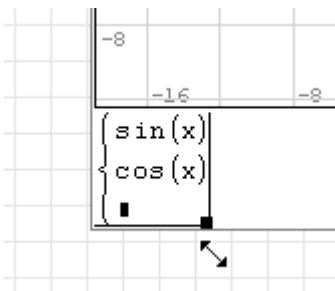
Slika 84. Grafici 8

Kada se približimo ovoj kockici kursor menja znak.



Slika 85. Grafici 9

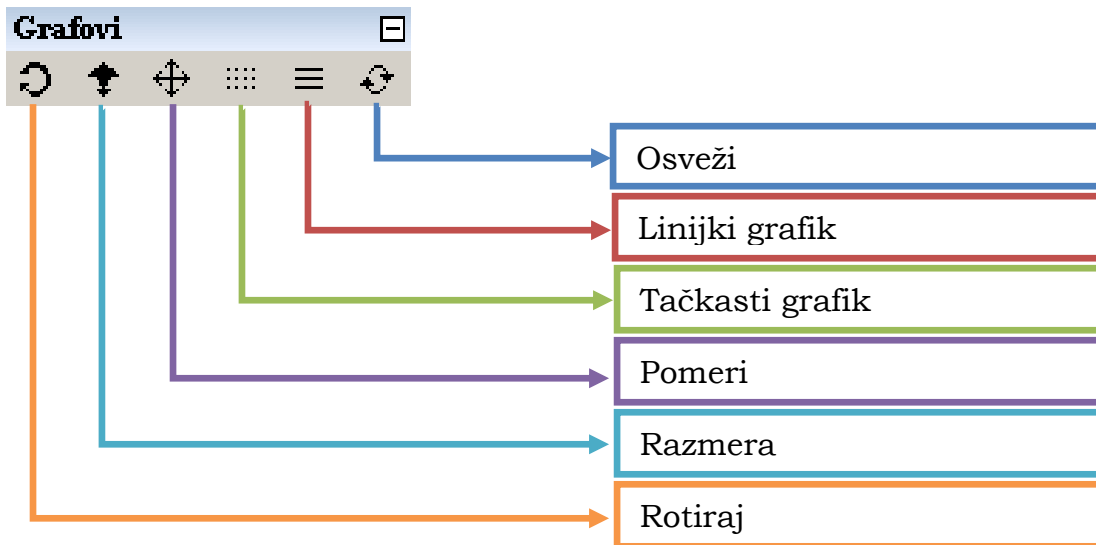
Pritiskom levog tastera miša i povlačenjem na dole ili gore povećavamo ili smanjujemo broj mesta.



Slika 86. Grafici 10

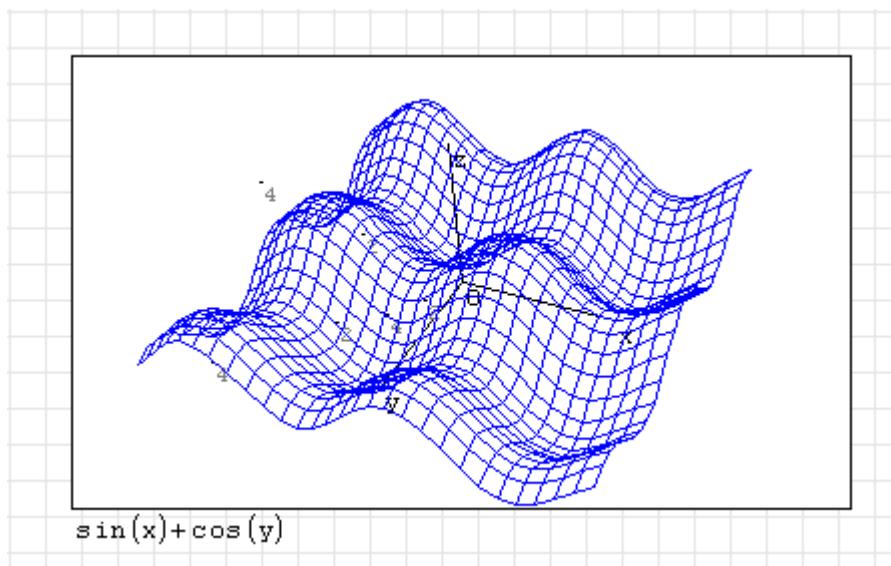
Puštanjem levog tastera završavamo ovu komandu.

Za manipulaciju sa graficima na raspolaganju su nam sledeće komande sa palete **Grafovi**.



Slika 87. Grafici 11

Primer 3D grafika



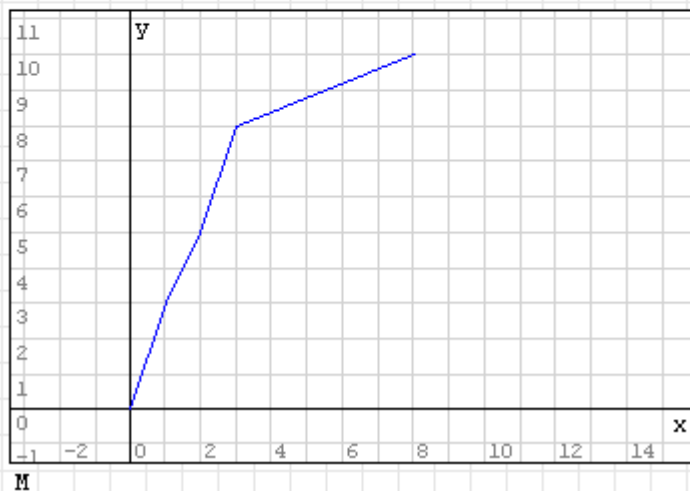
Slika 88. Grafici 12

Grafike možemo definisati i na drugi način, tako što ćemo umesto funkcije prikazati brojne vrednosti ordinata.

Primer:

$$M := \begin{pmatrix} 0 & 0 \\ 1 & 3 \\ 2 & 5 \\ 3 & 8 \\ 8 & 10 \end{pmatrix}$$

Definisana matrica M
Program prvu kolonu tretira kao
vrednosti apcise (x), a drugu kolonu
kao vrednosti ordinate (y)

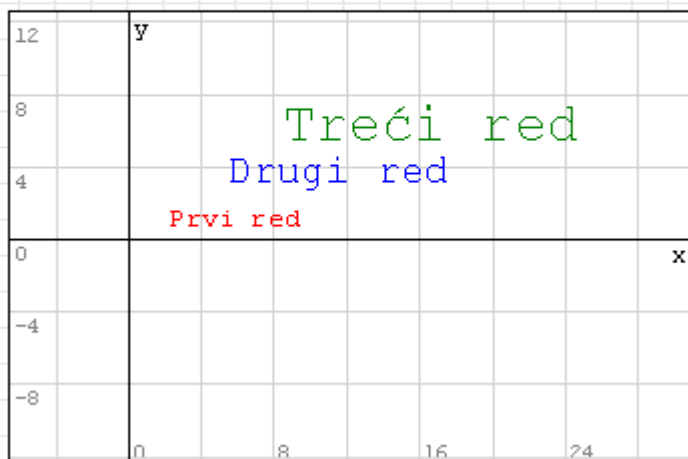


Slika 89. Grafici 13

Sledeći primer je primer ispisa teksta na grafiku

```
A := ( 2 2 "Prvi red" 10 "Red" )
      ( 5 5 "Drugi red" 15 "blue" )
      ( 8 8 "Treći red" 20 "green" )
```

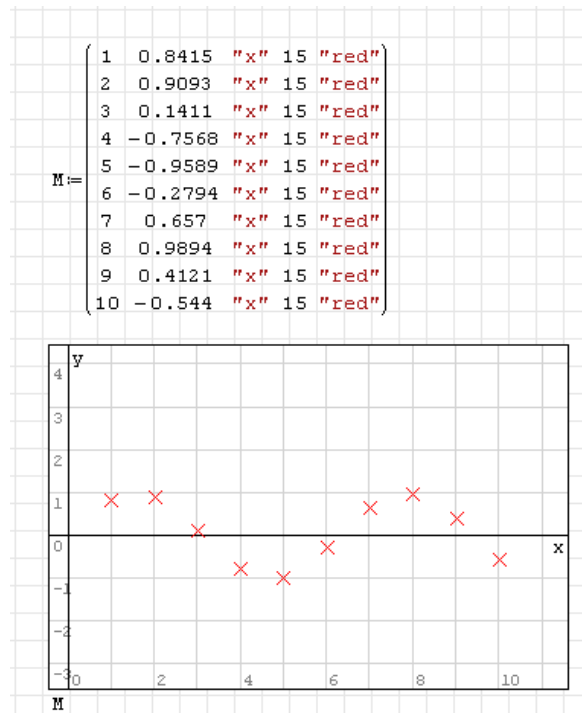
Objasnićemo na primeru
prvog reda.



Prve dve kolone predstavljaju x i y koordinatu, treća predstavlja tekst koji će biti ispisan, a preostale dve veličinu i boju teksta.

	x	y	tekst	veličina teksta	boja teksta
↑	↑	↑	↑	↑	↑
	2	2	"Prvi red"	10	"Red"

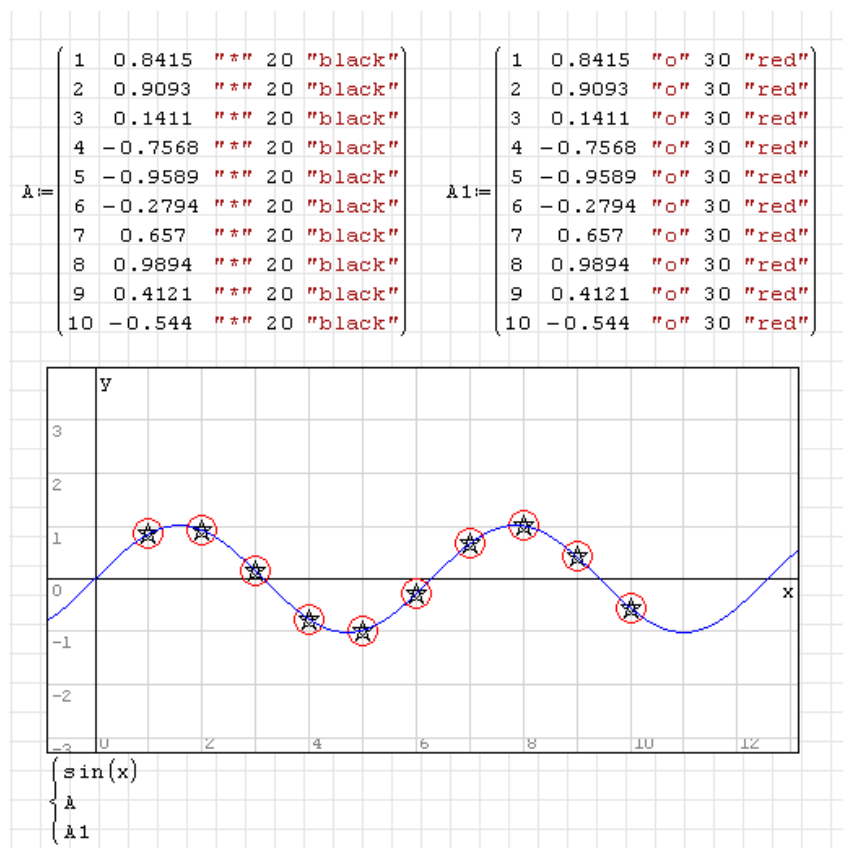
Slika 90. Grafici 14



Slika 91. Grafici 15

Na raspolaganju su nam sledeći znakovi : x , + , . , o , * itd.

Možemo raditi kombinaciju znakova i funkcija.



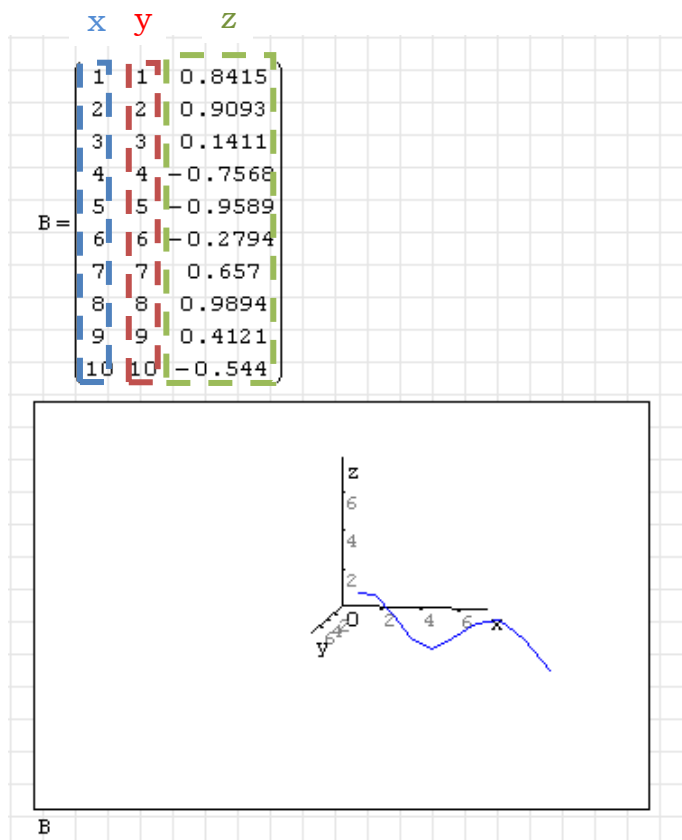
Slika 92. Grafici 16

Boje koje imamo na raspolaganju u graficima su sledeće:

Aliceblue, antiquewhite, aqua, aquamarine, azure, beige, blanchedalmond, blue, blueviolet, brown, burlywood, cadetblue, chartreuse, chocolate, coral, cornflowerblue, cornsilk, crimson, cyan, darkblue, darkcyan, darkgoldenrod, darkgray, darkgreen, darkhaki, darkmagenta, darkolivegreen, darkorange, darkorchid, darkred, darksalmon, darkseagreen, darkslateblue, darkslategray, darkturquoise, darkviolet, deeppink, deepskyblue, dimgray, dodgerblue, firebrick, floralwhite, forestgreen, fuchsia, gainsboro, ghostwhite, gold, goldenrod, gray, green, greenyellow, honeydew, hotpink, indianred, indigo, ivory, khaki, lavender, lavenderblush, lawngreen, lemonchiffon, lightblue, lightcoral, lightcyan, lightgoldenrodyellow, lightgray, lightgreen, lightpink, lightsalmon, lightseagreen, lightskyblue, lightslategray, lightsteelblue, lightyellow, lime, limegreen, linen, magenta, maroon, mediumaquamarine, mediumblue, mediumorchid, mediumpurple, mediumseagreen, mediumslateblue, mediumspringgreen, mediumturquoise, mediumvioletred, midnightblue, mintcream, mistyrose, moccasin, navajowhite, navy, oldlace, olive, olivedrab, orange, orangered, orchid, palegoldenrod, palegreen, paleturquoise, palevioletred, papayawhip, peachpuff, peru, pink, plum, powderblue, purple, red, rosybrown, royalblue, saddlebrown, salmon, sandybrown, seagreen, seashell, sienna, silver, skyblue, slateblue, slategray, snow, springgreen, steelblue, tan, teal, thistle, tomato, turquoise, violet, wheat, white, whitesmoke, yellow, yellowgreen, black.

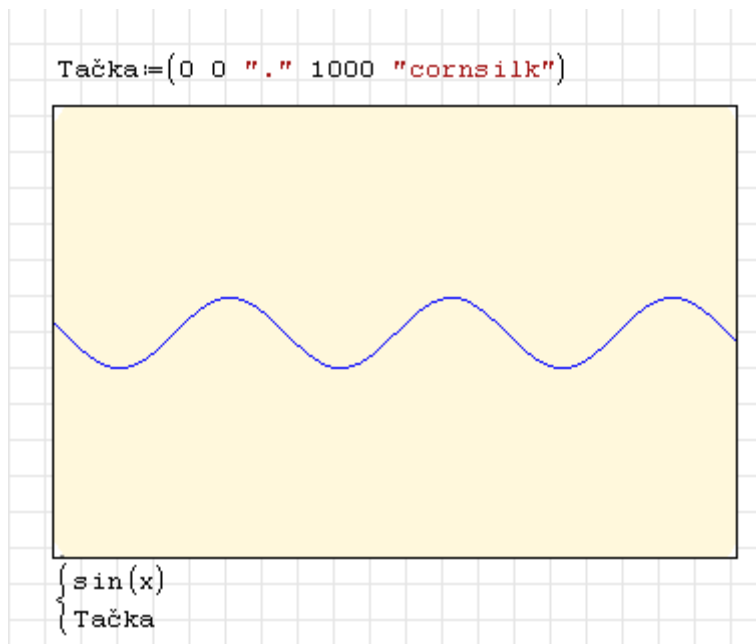
Ovi nazivi se mogu pisati malim ili velikim slovima (npr. darkgreen, DARKGREEN ili DarkGreen). Između naziva mogu se i staviti prazna mesta (npr. Dark Green).

Primer:



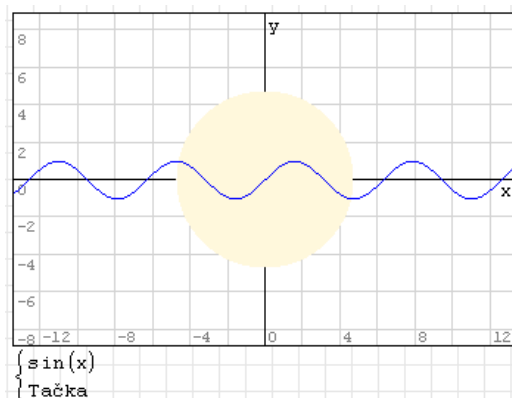
Slika 93. Grafici 17

Pošto u programu nije ostavljena mogućnost da u 2D prikazu promenimo boju pozadine, možemo iskoristiti činjenicu da je dozvoljeno postavljanje tačke "." proizvoljne veličine. Objašnjenje sledi u sledećem primeru.



Slika 94. Grafici 18

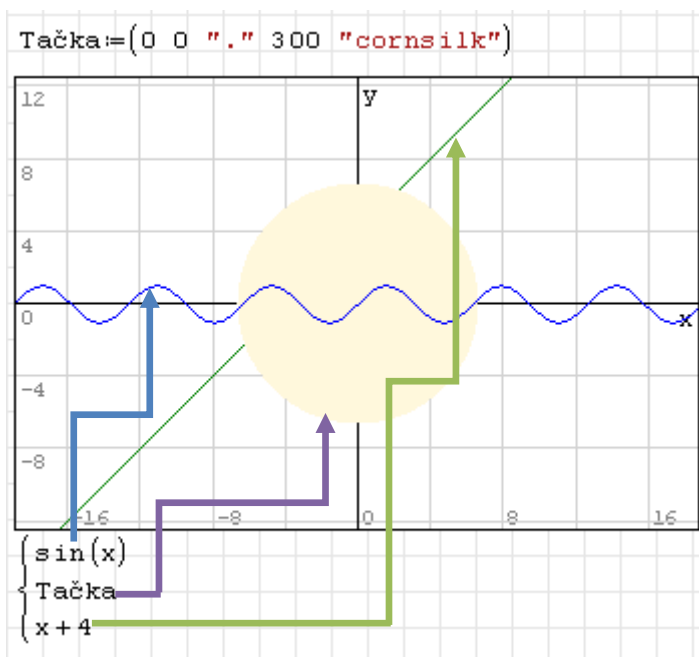
Iako izgleda da smo promenili boju pozadine grafika mi smo ustvari dodali jedan objekat (.) velike veličine (1000) koji je određene boje. Ako bismo dovoljno odzumirali videli bi sledeće.



Slika 95. Grafici 19

Važno je napomenuti da objekat treba da bude zadnji da se ne bi dobio prikaz kao na sledećoj slici. Naime može se uočiti da program prikazuje grafike po redosledu od gore ka dole.

Razmera osa x i y mogu se menjati kombinacijom tastera **Shift+točkić miša** i **Ctrl+točkić miša**.



Objekat **Tačka** se prikazuje iznad funkcije $x+4$ jer je takav redosled prikaza grafika.

Slika 96. Grafici 20

Rešavanje jednačina i nule funkcije

Primer:

Rešavanje kvadratne jednačine

$a=2$ $b=4$ $c=-6$

$$x = \frac{-b \pm \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a}$$

$$x = \begin{cases} 1 \\ -3 \end{cases}$$

$$x_1 = 1$$

$$x_2 = -3$$

kontrola

$2 \cdot 1^2 + 4 \cdot 1 - 6 = 0$
 $2 \cdot (-3)^2 + 4 \cdot (-3) - 6 = 0$

Do ovog izraza možemo doći kucajući sledeće:
 $x[1]=$ i $x[2]=$

Slika 97. Jednačine 1

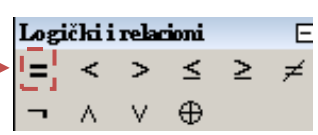
Ovo je moguće rešiti na ovaj način pošto nam je poznat analitički izraz za $x_{1,2}$. Ako ne poznajemo analitičko rešenje SMath studio poseduje nekoliko operacija a to su : **solve**, **roots** i **polyroots**.

Ako ukucamo sledeći izraz.

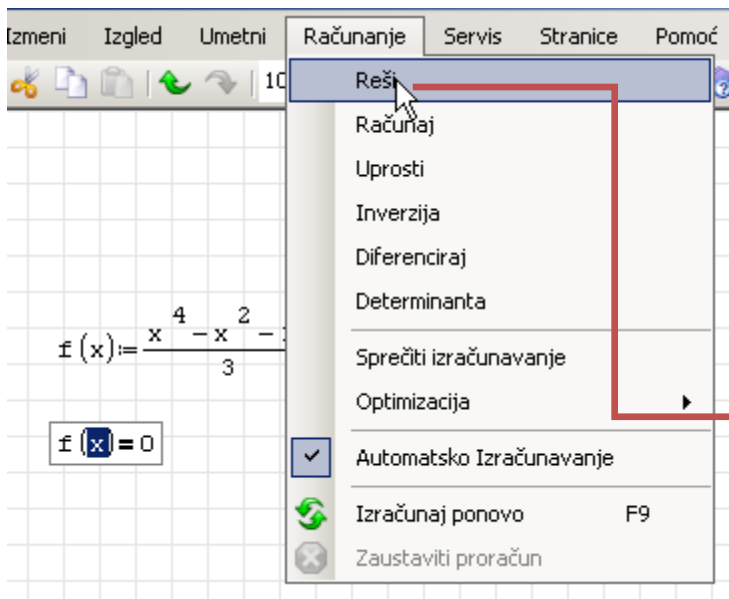
$$f(x) := \frac{x^4 - x^2 - 12}{3}$$

$$f(x) = 0$$

Primitite da je jednako sa palete **Logički i relacioni**



Slika 98. Jednačine 2



Ako obeležimo nepoznatu (x) i izaberemo komandu **Reši** sa padajućeg menija **Računanje** dobija se sledeće :

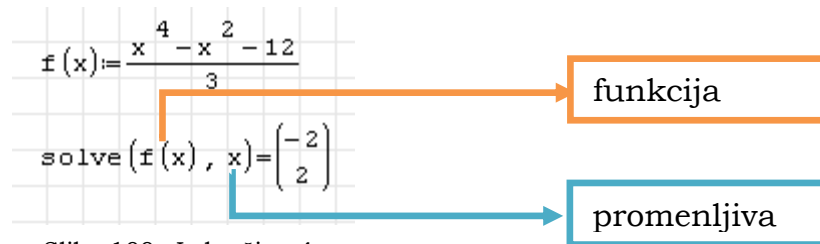
$$f(x) := \frac{x^4 - x^2 - 12}{3}$$

$$f(x) = 0$$

$$\begin{pmatrix} -2 \\ 2 \end{pmatrix}$$

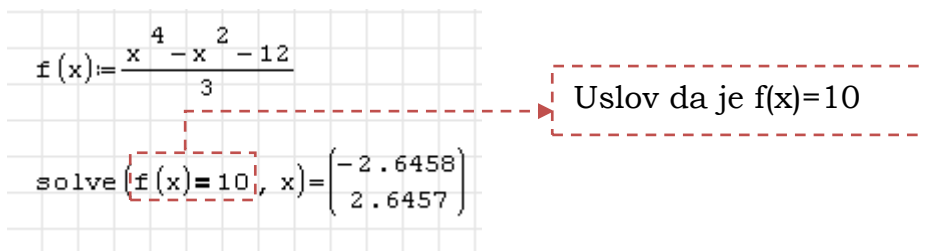
Slika 99. Jednačine 3

Ovo možemo rešiti i primenom funkcije **solve(2)**.



Slika 100. Jednačina 4

U gornjem primeru program računa kada je $f(x)=0$. Ovo možemo promeniti korišćenjem sledećeg izraza.

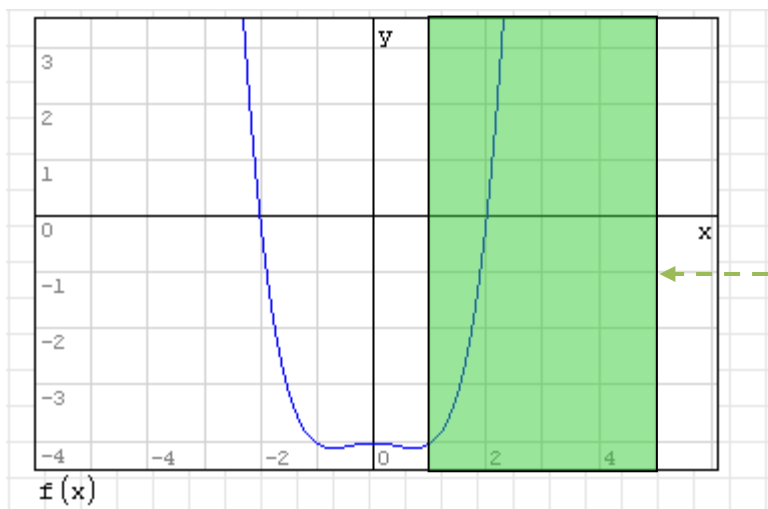


Slika 101. Jednačina 5

Prikaz funkcije **solve(4)** .

The diagram shows the function call `solve(f(x)=10, x, 1, 5)=2.6458` with three annotations:

- A red dashed box around `f(x)=10` is labeled "Uslov da je f(x)=10".
- A green dashed box around `x, 1, 5` is labeled "Opseg promenljive".
- A blue dashed box around `x` is labeled "Promenljiva".



Slika 102. Jednačina 6

Funkcija **roots(2)** prikazuje prvu vrednost nule funkcije.

The diagram shows the function call `roots(f(x), x)=-2` and the function definition `f(x) = (x^4 - x^2 - 12) / 3`. A text box explains: "U ovom slučaju to je vrednost -2 što je vidljivo iz gornjeg grafika funkcija."

Slika 103. Jednačina 7

Funkcija **roots(3)** ima jedan dodatni argument koji služi kao pretpostavka rešenja.

The diagram shows the function call `roots(f(x), x, 3)=2` and `roots(f(x), x, -1)=-2` with three annotations:

- A blue dashed box around `3` is labeled "Pretpostavka".
- A red dashed box around `x` is labeled "Promenljiva".
- A green dashed box around `f(x)` is labeled "Funkcija".

Slika 104. Jednačina 8

Polyroots funkcija nam služi da tražimo nule polinoma. Osim realnih nula koje daju prethodne komande ova nam daje i imaginarne vrednosti.

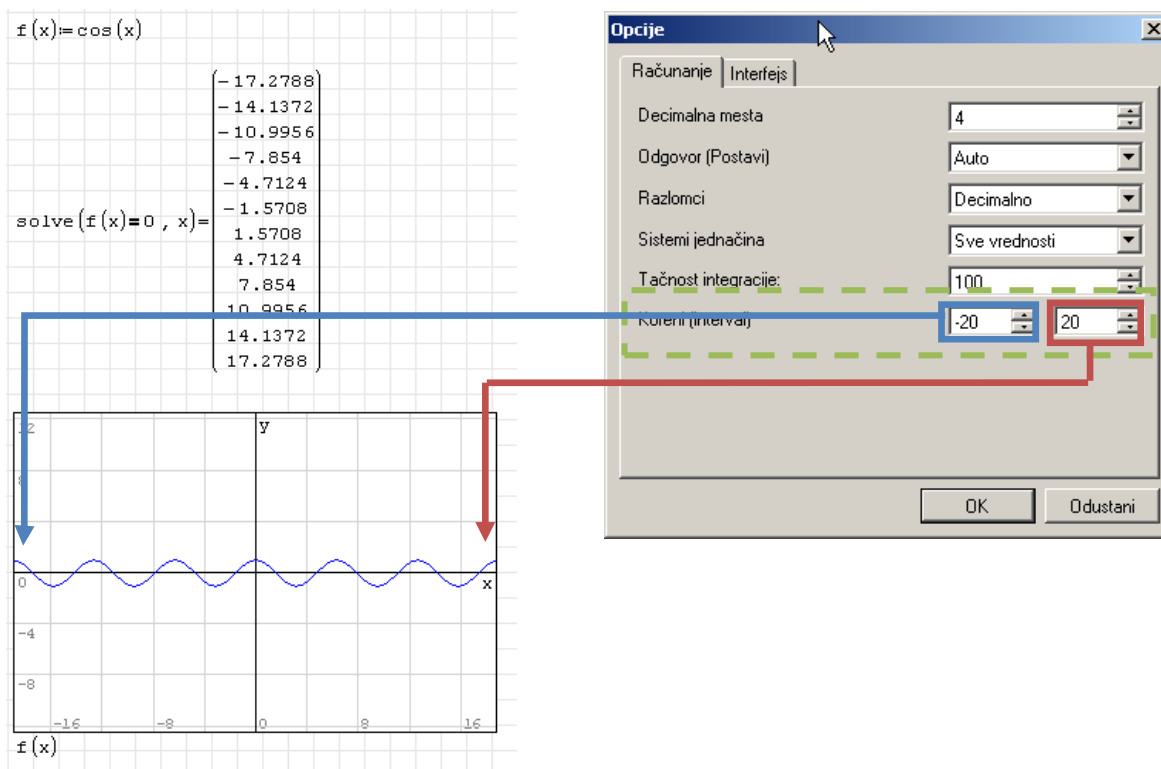
$$f(x) := \frac{x^4 - x^2 - 12}{3}$$

$$f(x) := \frac{1}{3} \cdot x^4 - \frac{1}{3} \cdot x^2 - 4$$

$$\text{polyroots} \left(\begin{pmatrix} -4 \\ 0 \\ -\frac{1}{3} \\ 0 \\ \frac{1}{3} \end{pmatrix} \right) = \begin{pmatrix} 1.7321 \cdot i \\ -1.7321 \cdot i \\ -2 \\ 2 \end{pmatrix}$$

Slika 105. Jednačina 9

Evo jednog primera koji će nam objasniti interval tj. granice.



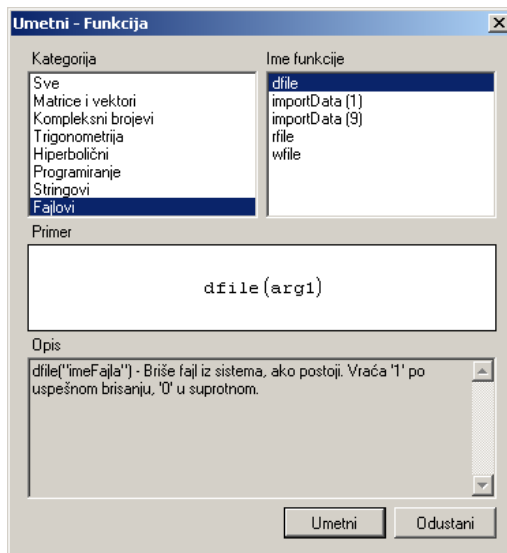
Slika 106. Jednačina 10

Iako trigonometrijska funkcija ima puno nula program je prikazao rezultate u granicama od -20 do +20. Granice se mogu promeniti izborom iz padajućeg menija **Servis/Opcije**.

Ili korišćenjem komande **solve(4)** i postavljanje većeg/manjeg intervala npr. `solve(f(x) = 0, x, -30,30)`

Pisanje i čitanje podataka iz fajla

Smath Studio poseduje set komandi koje služe da podatke upisujemo i čitamo iz fajla. Do ovih funkcija može se doći izborom iz padajućeg menija **Umetni / Funkcije**, a zatim izborom iz grupe **Fajlovi**.

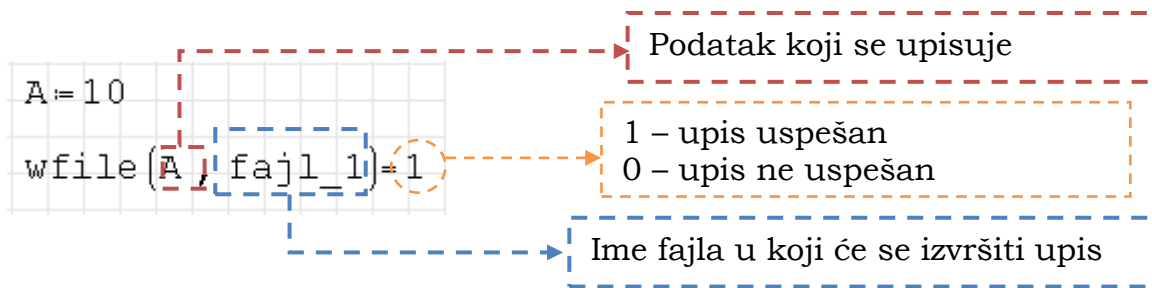


Slika 107. Podatci 1

Program sve fajlove koje čita i snima čuva u **C:\Documents and Settings\[IME KORISNIKA]\Application Data\SMath\user** (U pitanju je Windows XP)

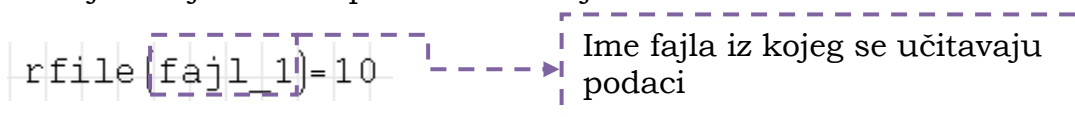
Ovaj folder je skriven tako da ga ne možete direktno naći. Ovo možete rešiti tako što iz Windows-ovog prozora iz padajućeg menija **Tools/Folder Options** na kartici **View** otkačite **Show hidden files and folders**.

Primer upisa podataka u fajl prikazaćemo na sledećem primeru.



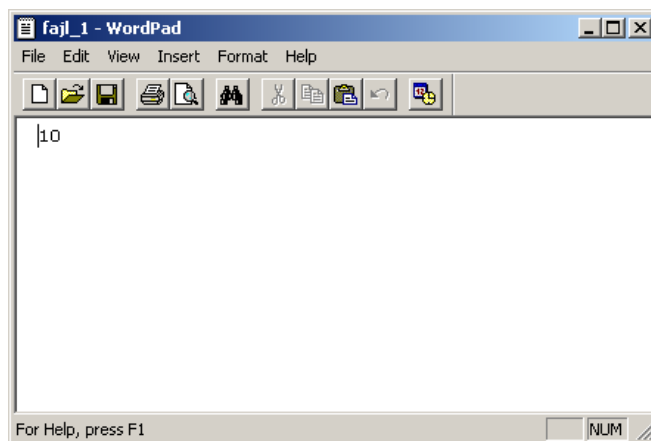
Slika 108. Podaci 2

Čitanje iz fajla se vrši pomoću funkcije **rfile**.



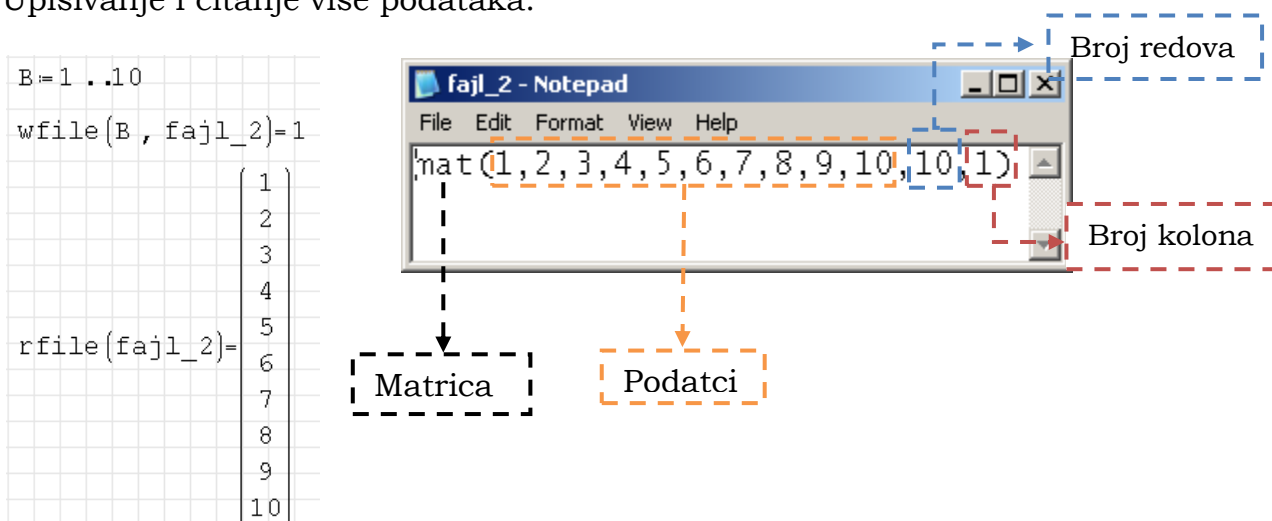
Slika 109. Podaci 3

Same fajlove možemo otvoriti sa programima tipa Notepad, Wordpad.



Slika 110. Podaci 4

Upisivanje i čitanje više podataka.



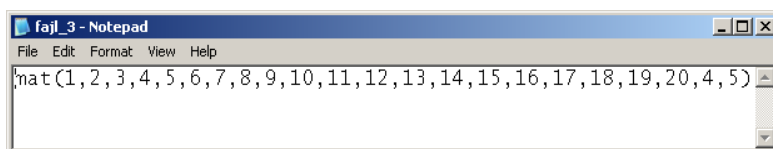
Slika 111. Podaci 5

```

C =
( 1 2 3 4 5
 6 7 8 9 10
11 12 13 14 15
16 17 18 19 20)

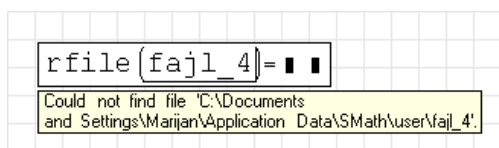
wfile(C, fajl_3)=1

rfile(fajl_3)=
( 1 2 3 4 5
 6 7 8 9 10
11 12 13 14 15
16 17 18 19 20)
    
```



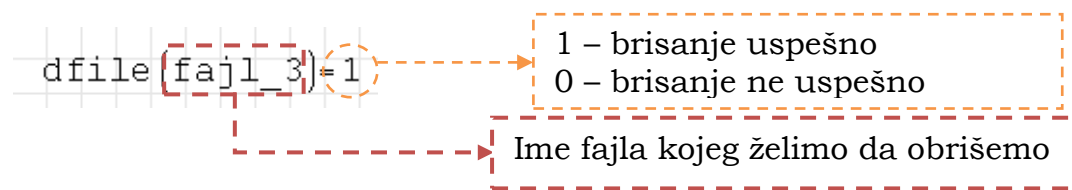
Slika 112. Podaci 6

U slučaju da želimo da pročitamo fajl kojeg nema program će ispisati sledeće obaveštenje.



Slika 113. Podatci 7

Brisanje fajla iz sistema se vrši pozivanjem komande **dfile**.



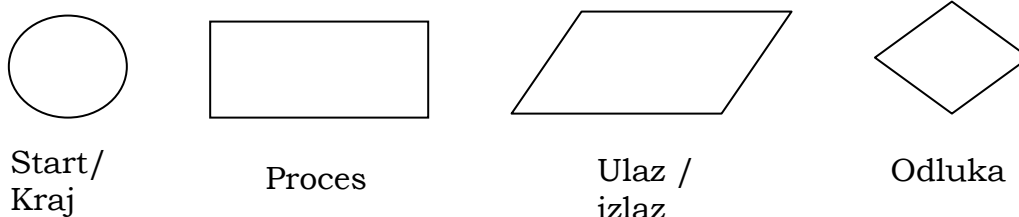
Slika 114. Podaci 8

Programiranje

I ovu oblast ću pokušati objasniti u duhu ovog priručnika. Pretpostaviću da o programiranju znate apsolutno ništa.

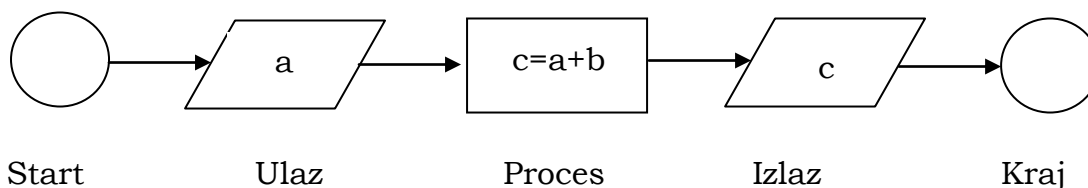
Cela suština je da programu (kompjuteru) kažete da uradi neke operacije po nekom redosledu i da Vam prikaže rezultate. Možda je bolja formulacija da program uzme ulazne podatke, da ih obradi i prikaže izlazne.

Da bi smo u programu napisali šta treba da se uradi moramo rešiti određene korake. Njih ćemo prikazati grafički.



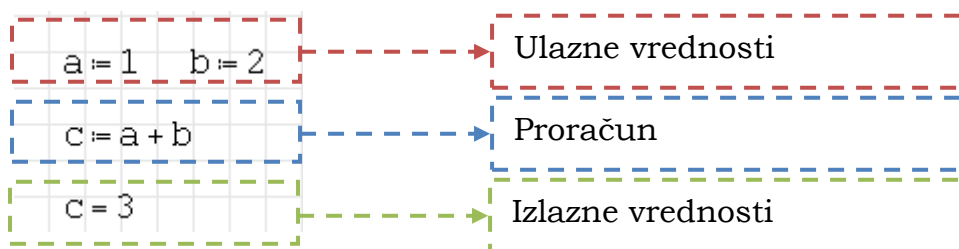
Slika 115. Programiranje 1

Sada će biti prikazan jedan običan algoritam. Naime za ulazne vrednosti **a** i **b** program će izračunati vrednost **c** koja je jednaka zbiru **a** i **b**.



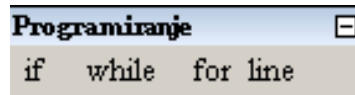
Slika 116. Programiranje 2

Kada bi ovo hteli da zapišemo u programu SMath Studio izraz bi izgledao ovako:



Slika 117. Programiranje 3

SMath poseduje paletu **Programiranje** na kojoj se nalaze komande koje služe pri programiranju.



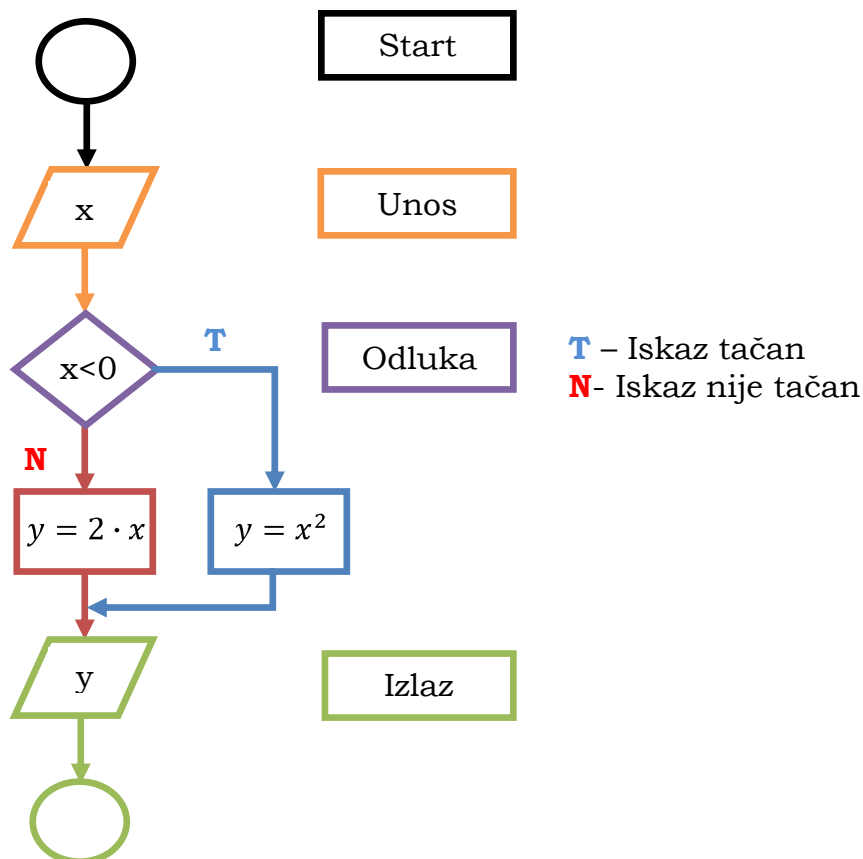
Slika 118. Programiranje 4

IF komanda

If komanda nam služi ako imamo granjanje. Da bi smo ovo objasnili poslužiće nam sledeći primer. Ako imamo neku funkciju koja ima oblik.

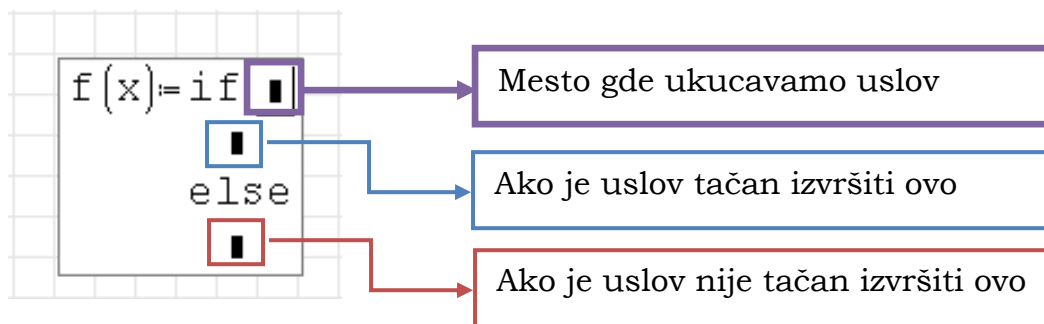
$$f(x) = \begin{cases} x^2, & x < 0 \\ 2 \cdot x, & x \geq 0 \end{cases}$$

Ako bismo za ovako definisanu funkciju hteli da napišemo program koji za vrednost promenljive x daje rezultat funkcije $f(x)$ morali bismo da napravimo granjanje. Algoritam za ovaj problem bi izgledao sledeće.



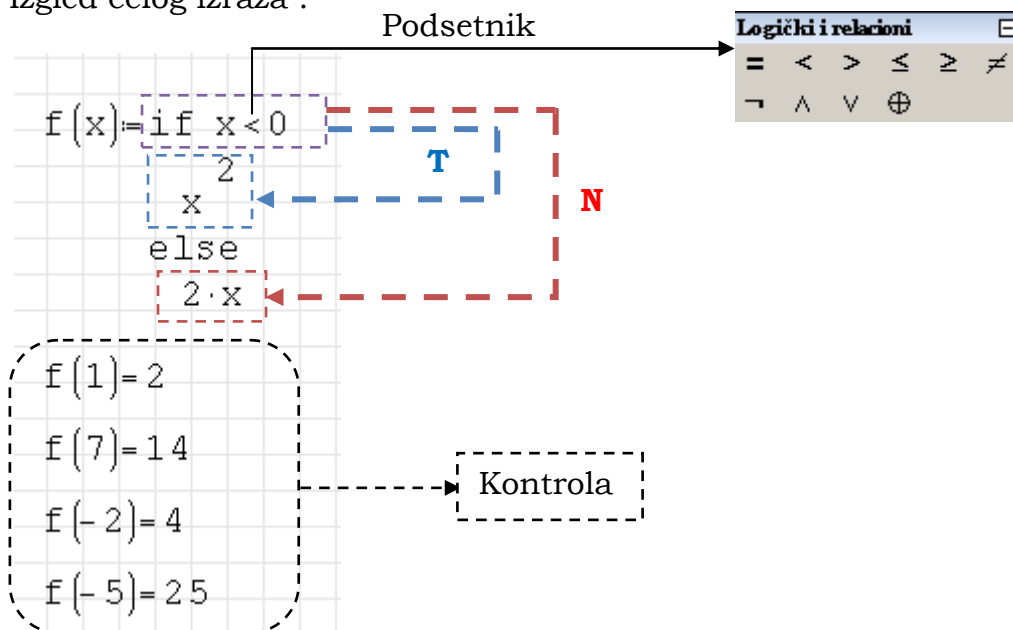
Slika 119. Programiranje 5

Kako bi ovaj program uneli u SMath. Upisali bi sledeće $f(x)$: ,a onda bi sa palete **Programiranje** pritisnuli komandu **if**. Prikaz na radnom listu je sledeći :



Slika 120. Programiranje 6

Izgled celog izraza :



Slika 121. Programiranje 7

Vidimo da u ovom primeru postoji logički uslov koji treba da bude ispunjen li ne ispunjen ($x < 0$) da bi program uzeo jednu ili drugu vrednost

$$f(x) = x^2 \text{ ili } f(x) = 2x$$

U ovom trenutku mogli bismo se upoznati sa paletom **Logički i relacioni**.



Slika 122. Programiranje 8

Mislim da prvih pet nije potrebno dodatno objašnjavati (jednako, manje, veće, manje ili jednako, veće ili jednako i nije jednako).

Primer:

$$3 > 2 = 1 \text{ (iskaz je tačan)}$$

$$3 = 2 = 0 \text{ (iskaz nije tačan)}$$

$$5 \geq \pi = 1 \text{ (iskaz je tačan)}$$

$$3 < 2 = 0 \text{ (iskaz nije tačan)}$$

$$3 \neq 2 = 1 \text{ (iskaz je tačan)}$$

$$5 \leq \pi = 0 \text{ (iskaz nije tačan)}$$

\neg Negacija (ne)

$$\neg 1 = 0$$

$$\neg 0 = 1$$

\wedge Konjunkcija (i)

$$1 \wedge 1 = 1 \text{ (jedan i drugi izraz su tačni = ceo izraz je tačan)}$$

$$1 \wedge 0 = 0$$

$$0 \wedge 1 = 0$$

$$0 \wedge 0 = 0$$

\vee Disjunkcija (ili)

$$1 \vee 1 = 1 \text{ (prvi ili drugi izraz je tačan = ceo izraz je tačan)}$$

$$1 \vee 0 = 1$$

$$0 \vee 1 = 1$$

$$0 \vee 0 = 0$$

\oplus Ekskluzivna disjunkcija

$$1 \oplus 1 = 0 \text{ (Tačno je jedino ako je samo jedan izraz tačan)}$$

$$1 \oplus 0 = 1$$

$$0 \oplus 1 = 1$$

$$0 \oplus 0 = 0$$

Logičke operacije

Negacija (ne)

$$\neg (3 > 2) = 0$$

Konjunkcija (i)

$$(3 > 2) \wedge (4 > 3) = 1$$

Disjunkcija (ili)

$$(3 > 2) \vee (5 < 4) = 1$$

Ekskluzivna disjunkcija

$$(3 < 2) \oplus (2 > 1) = 1$$

Kod funkcije **if** u programu postoji mali bug.

```
f(x) := if x = 1.1
      10
      else
      9
f(1.1) = 9
```

Tačno rešenje
je 10

Slika 123. Programiranje 9

Ovo možemo rešiti opet na nekoliko načina. Prvi je da brojeve napišemo kao razlomke.

```
f(x) := if x =  $\frac{11}{10}$ 
        10
        else
        9

f( $\frac{11}{10}$ ) = 10
```

Slika 124. Programiranje 10

Drugi način je da iskoristimo funkciju **Line** (Dodaj red). Ako napišemo $f(x):$,a zatim pritisnemo komandu **Line** na radnom listu će biti prikazano sledeće :

```
f(x) := |
```

Slika 125. Programiranje 11

U prvi plejsholder upišemo izraz, a zatim tipkom space ili strelicama obuhvatimo ceo izraz.

```
f(x) := if x = 1.1
        10
        else
        9
```

Slika 126. Programiranje 12

U levom uglu pojaviće se crna kockica. Kada dovedemo pokazivač miša do nje, on menja oblik. Levim klikom hvatamo i smanjujemo ceo izraz.

```
f(x) := if x = 1.1
        10
        else
        9
```

Slika 127. Programiranje 13

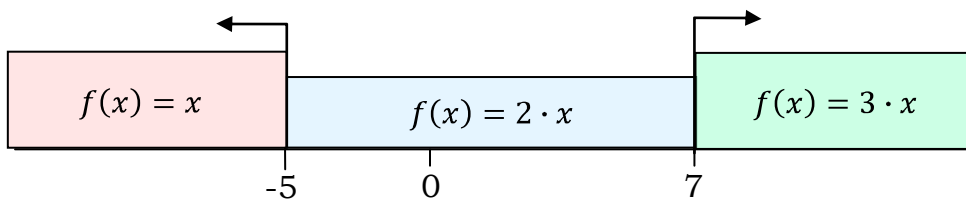
Pogledajmo sad kompletan izraz.

```
f(x):= | if x = 1.1
      | 10
      | else
      | 9
      |
      | f(1.1) = 10
```

Slika 128. Programiranje 14

Primer:

Šematski je prikazana funkcija tj. opsezi u kojima funkcija uzima određene vrednosti.



Slika 129. Programiranje 15

Zapis u SMath Studiu :

```
f(x):= | if x <= -5
      | x
      | else
      | if x >= 7
      | 3 * x
      | else
      | 2 * x
```

Slika 130. Programiranje 16

Ovo se u programu unosi na sledeći način. Prvo se unese kao što je prikazano na sledećoj slici.

```
f(x):= | if
      |
      | else
      |
```

Slika 131. Programiranje 17

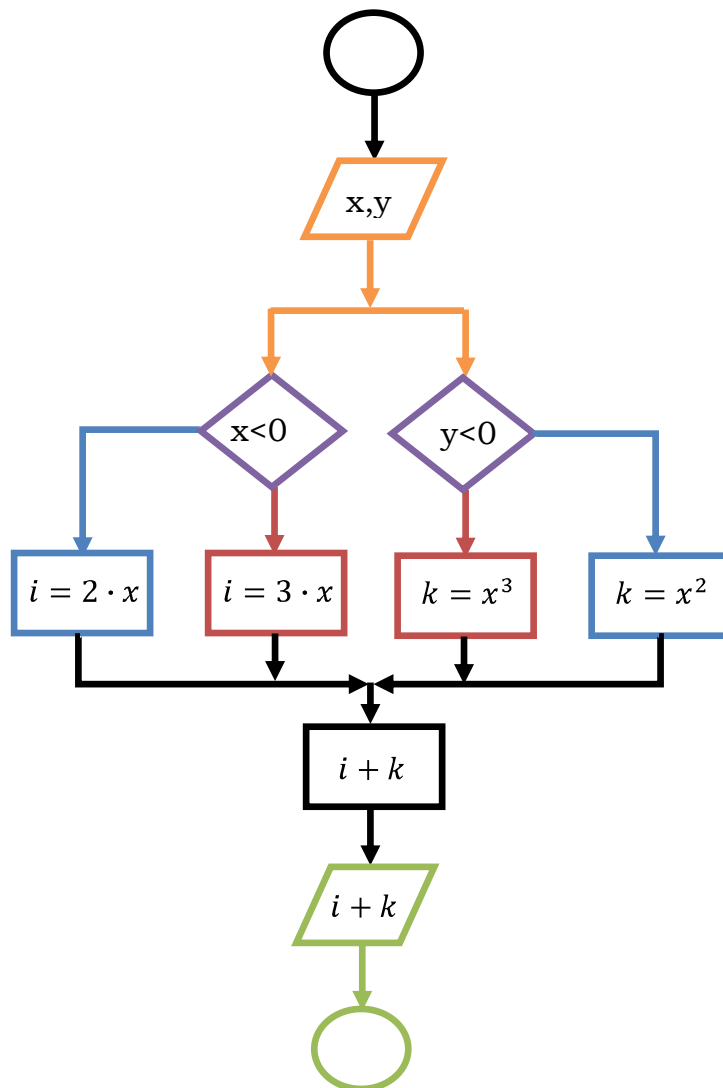
Zatim dovedemo liniju za unos u kvadratić ispod **else** i dodamo još jednu **if** funkciju.


```
f(x) := if
      |
      |
      | else
      |   if
      |   |
      |   | else
      |   |
      |   |
      |   |
```

Slika 132. Programiranje 18

Poslednji plejsholder “obrišemo” na isti način kako je objašnjeno na slikama 126 i 127 .

Analizirajmo sledeći algoritam. U zavisnosti od ulaznih parametara **x** i **y** program će izračunati zavisne promenljive **i** i **k** i zatim ih sabrati i prikazati rezultat .



Slika 133. Programiranje 14

```
f(x, y) := | if x < 0
            |   i := 2 · x
            | else
            |   i := 3 · x
            | if y < 0
            |   k := y2
            | else
            |   k := y3
            | i + k
f(1, 1) = 4
f(-5, -3) = -1
```

Slika 134. Programiranje 15

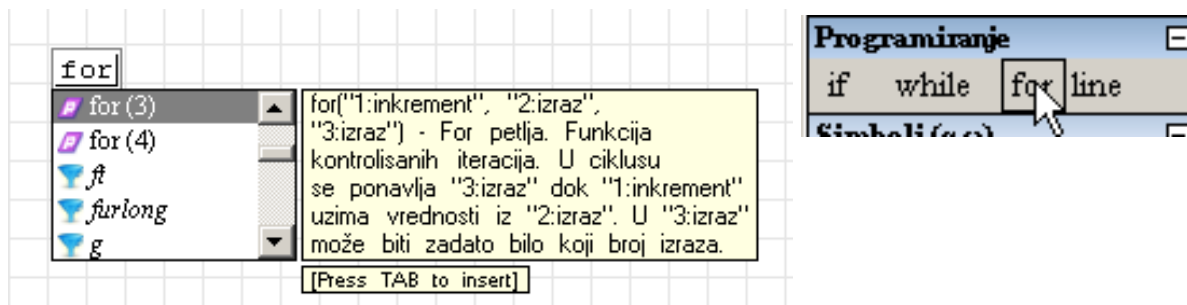
Primetimo sledeće, program ispisuje samo rezultat. Ako bismo hteli da program kao izlazne vrednosti osim rezultata ispiše i vrednosti za promenljive **i** i **k** morali bi da zadnju liniju koda stavimo jedan vektor sa tri reda i jednom kolonom.

```
f(x, y) := | if x < 0
            |   i := 2 · x
            | else
            |   i := 3 · x
            | if y < 0
            |   k := y2
            | else
            |   k := y3
            | ( i + k
              | i
              | k )
f(1, 1) = ( 4
           | 3
           | 1 )
```

Slika 135. Programiranje 16

FOR petlja

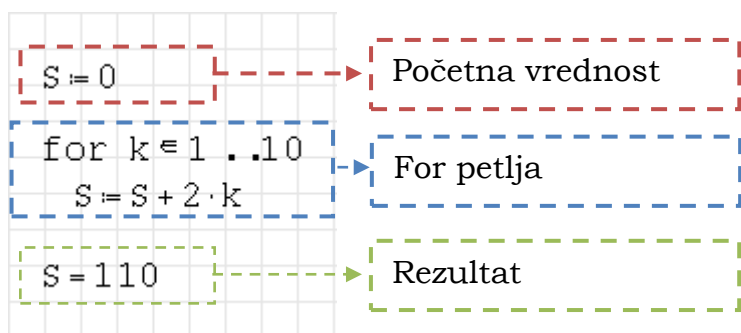
Ova vrsta petlji nam služi ako želimo da deo programa ponovimo određeni broj puta. SMath Studio poseduje dve vrste **for** petlji **for(3)** i **for(4)**. Ovu funkciju u program možemo unositi na više načina. Kucajući tekst **for** i izborom preko dinamičkog asistenta ili pritiskom na paleti.



Slika 136. For petlja 1

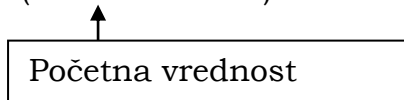
Ove kao i naredne vrste petlji objasnićemo kroz primere.

Primer :

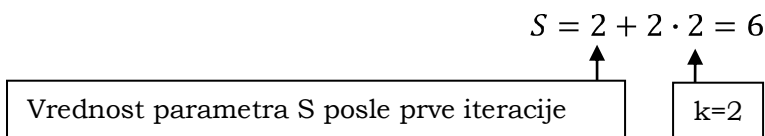


Slika 137. For petlja 2

Analizirajmo ovu petlju. Na početku smo parametru S zadali početnu vrednost ($S=0$). Zatim dolazimo do petlje gde parametar k uzima vrednost jedan ($k=1$) i izvršava se proračun $S = S + 2 \cdot k$ ($S = 0 + 2 \cdot 1 = 2$)



Posle prve iteracije parametar S sada ima vrednost dva. Petlja se ponovo izvršava za vrednost $k=2$.



Parametar S sada ima novu vrednost $S=6$, k uzima novu vrednost $k=3$ i petlja se ponovo izvršava. Ovo se ponavlja do vrednosti $k=10$.

Ceo proračun može da se prikaže na sledeći način.

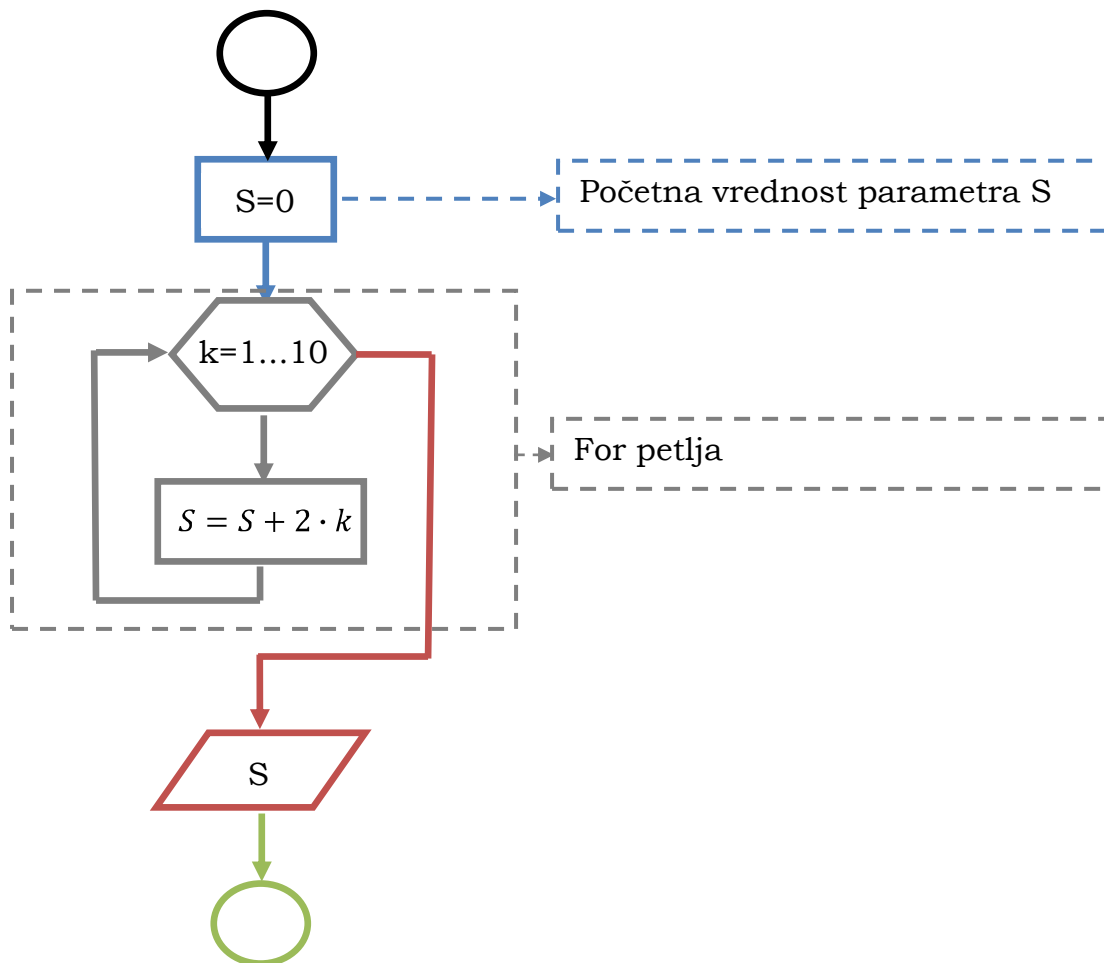
```

    2+4+6+8+10+12+14+16+18+20=110
    ┌
k=1;S=2
    └
    ┌
k=2;S=2+4=6
    └
    ┌
k=2;S=2+4+6=10
    └
    ┌──────────────────────────┐
k=10;S=2+4+6+8+10+...+20=110
    └──────────────────────────┘
    
```

Matematički ovaj primer može da se zapiše na sledeći način.

$$S = \sum_{k=1}^{10} 2 \cdot k = 110$$

Algoritam proračuna



Slika 138. For petlja 3

Analizirajmo sledeći problem

$$\sum_{j=1}^5 \sum_{k=1}^5 (k \cdot j) = 225$$

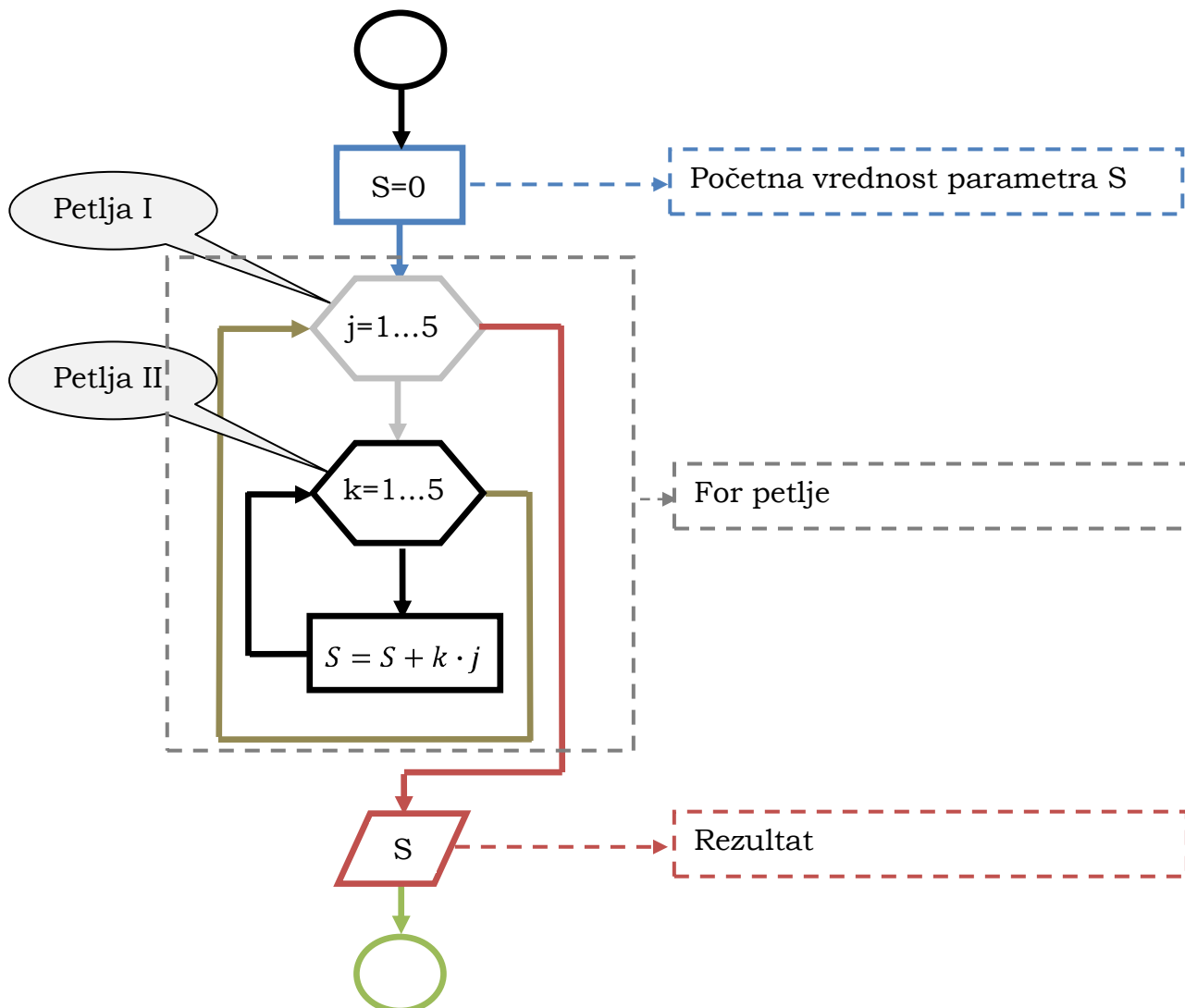
Rešenje ovog problema primenom for petlji je sledeći.

```

S = 0
for j ∈ 1 ..5
  for k ∈ 1 ..5
    S = S + k · j
S = 225
    
```

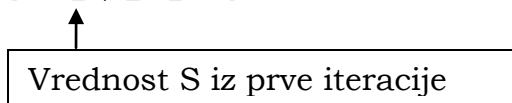
Slika 139. For petlja 4

Probaću da na što jednostavniji način objasnim ove petlje. U tome će mi pomoći algoritam rešavanja.



Slika 140. For petlja 4

Sam redosled izvršavanje ovog koda tekao bi na sledeći način. Iz prve petlje bi se uzela vrednost $j=1$, a iz druge $k=1$. Rezultat je $(S = S + k \cdot j) S = 0 + 1 \cdot 1 = 1$. Program se vraća u drugu petlju i uzima novu vrednost za parametar $k=2$.
 $S = 1 + 2 \cdot 1 = 3$



Proračun bi se izvršavao za sve vrednosti parametra k (1,2,3,4 i 5). Kada “izvrtimo” sve vrednosti iz druge petlje, program se vraća u prvu petlju i uzima vrednost $j=2$ i ponavlja ceo proračun.

Prikaz rešavanja

za $j=1$ $0+1\cdot1+2\cdot1+3\cdot1+4\cdot1+5\cdot1+$
 za $j=2$ $+1\cdot2+2\cdot2+3\cdot2+4\cdot2+5\cdot2+$
 za $j=3$ $+1\cdot3+2\cdot3+3\cdot3+4\cdot3+5\cdot3+$
 za $j=4$ $+1\cdot4+2\cdot4+3\cdot4+4\cdot4+5\cdot4+$
 za $j=5$ $+1\cdot5+2\cdot5+3\cdot5+4\cdot5+5\cdot5=$ **225**

Crvenom bojom su obeleženi brojevi koji odgovaraju parametru k

Ovu petlju smo u SMath Studiu mogli zapisati i na sledeći način.

```
S := | S = 0
      | for j ∈ 1 .. 5
      |   for k ∈ 1 .. 5
      |     S = S + k · j
      | S
S = 225
```

Slika 141. For petlja 5

Razlika u ovom zapisu i zapisu ranije (Slika 139) je u tome što je u ovom slučaju ceo kod zapisan u liniji tako da su svi parametri (k i j) lokalne promenljive i njihove vrednosti nisu dostupne izvan programa. Naime ako bi u prvom slučaju tražili vrednost za k ili j program bi ih ispisao (zadnje vrednosti iz petlje), dok u drugom zapisu to ne bi bilo moguće pošto su lokalne promenljive.

```
S = 0

for j ∈ 1 .. 5
  for k ∈ 1 .. 5
    S = S + k · j

S = 225
j = 5
k = 5
```

Slika 142. For petlja 6

Sledeći primer kombinuje tri programske strukture, dve **for** petlje i jednu **if** funkciju. Program sortira podatke od najmanjeg do najvećeg i prikazuje ih u vidu matrice .

Program napisan u SMath Studiu:

```

rS:=(5.4 1.2 3.5 10.2 -2.5 4.1)
ns:=length(rS)      ns=6

for k∈1..ns-1
  for j∈k+1..ns
    if rS1 k>rS1 j
      temp:=rS1 j
      rS1 j:=rS1 k
      rS1 k:=temp
    else
      0
rS=(-2.5 1.2 3.5 4.1 5.4 10.2)

```

Slika 143. For petlja 7

Postupno će biti objašnjen ceo proces proračuna.

Komandom **length()** određujemo broj elemenata u matrici rS. Ovu vrednost dodeljujemo promenljivoj **ns** koja je jednaka broju šest. Sada dolazimo do petlje I, vrednost promenljive k=1. Petlja II vrednost j=k+1=1+1=2. U sledećem koraku program poredi vrednosti $rS_{11} > rS_{12}$ tj. da li je $5.4 > 1.2$, pošto je tvrdnja tačna program se nastavlja.

$$\begin{aligned}
 temp &= rS_{12} ; temp = 1.2 \\
 rS_{12} &= rS_{11} ; rS_{12} = 5.4 \\
 rS_{11} &= temp ; rS_{11} = 1.2
 \end{aligned}$$

Ako bi sada pogledali ovu matricu posle prve iteracija ona bi izgledala:

$$rS^{\text{posle prve iteracije}} = (1.2 \ 5.4 \ 3.5 \ 10.2 \ -2.5 \ 4.1)$$

Ponovo se vraćamo na petlju II, sada je j=3 (k=1).

Kontrola $rS_{11} > rS_{13}$ tj. $1.2 > 3.5$ Pošto tvrdnja nije tačna petlja završava svoj ciklus.

j=4 (k=1) ; $rS_{11} > rS_{14}$ ($1.2 > 10.2$) – kraj ciklusa

j=5 (k=1) ; $rS_{11} > rS_{15}$ ($1.2 > -2.5$) – tvrdnja je tačna i program se nastavlja

$$\begin{aligned}
 temp &= rS_{15} ; temp = -2.5 \\
 rS_{15} &= rS_{11} ; rS_{15} = 1.2 \\
 rS_{11} &= temp ; rS_{11} = -2.5
 \end{aligned}$$

j=6 (k=1) ; $rS_{11} > rS_{16}$ ($-2.5 > 4.1$) – kraj ciklusa

Pošto je petlja II ($j=2,3,4,5,6$) "izvrtela" sve vrednosti program se vraća na petlju I i uzima vrednost za $k=2$ i ponovo vraća na petlju II gde j uzima vrednost $k+1$.
 $k=2$; $j=k+1=2+1=3$; $rS_{12} > rS_{13}$ ($5.4 > 3.5$) – tvrdnja je tačna

$$\begin{aligned} temp &= rS_{13} ; temp = 3.5 \\ rS_{13} &= rS_{12} ; rS_{13} = 5.4 \\ rS_{12} &= temp ; rS_{12} = 3.5 \end{aligned}$$

Možemo da zaključimo da su vrednosti 5.4 i 3.5 zamenili mesta . Sada matrica rS izgleda:

$$(-2.5 \ 3.5 \ 5.4 \ 10.2 \ 1.2 \ 4.1)$$

Kada ponovimo ceo ciklus za vrednosti $k=2$ i $j=3,4,5$ matrica rS izgleda :

$$(-2.5 \ 1.2 \ 5.4 \ 10.2 \ 3.5 \ 4.1)$$

Po završetku svih ciklusa $k=1,2,3,4,5$ i $j=2,3,4,5,6$ elementi matrice rS biće poređani na sledeći način.

$$rS = (-2.5 \ 1.2 \ 3.5 \ 4.1 \ 5.4 \ 10.2)$$

Od ovog algoritma možemo napraviti funkciju koju zatim možemo pozivati svaki put kada želimo da sortiramo vrednosti.

```

mySort(rS)=| ns=length(rS)
            | for k∈1..ns-1
            |   for j∈k+1..ns
            |     if rS_1 k > rS_1 j
            |       temp:=rS_1 j
            |       rS_1 j:=rS_1 k
            |       rS_1 k:=temp
            |     else
            |       0
            | rS

A=(5.4 1.2 3.5 10.2 -2.5 4.1)
B=(0 -2.65 -3.45 10)
mySort(A)=(-2.5 1.2 3.5 4.1 5.4 10.2)
mySort(B)=(-3.45 -2.65 0 10)
    
```

Slika 144. For petlja 8

Do istog rešenje mogli smo doći primenom funkcije **sort()** koja sortira elemente u uzlaznom redosledu.

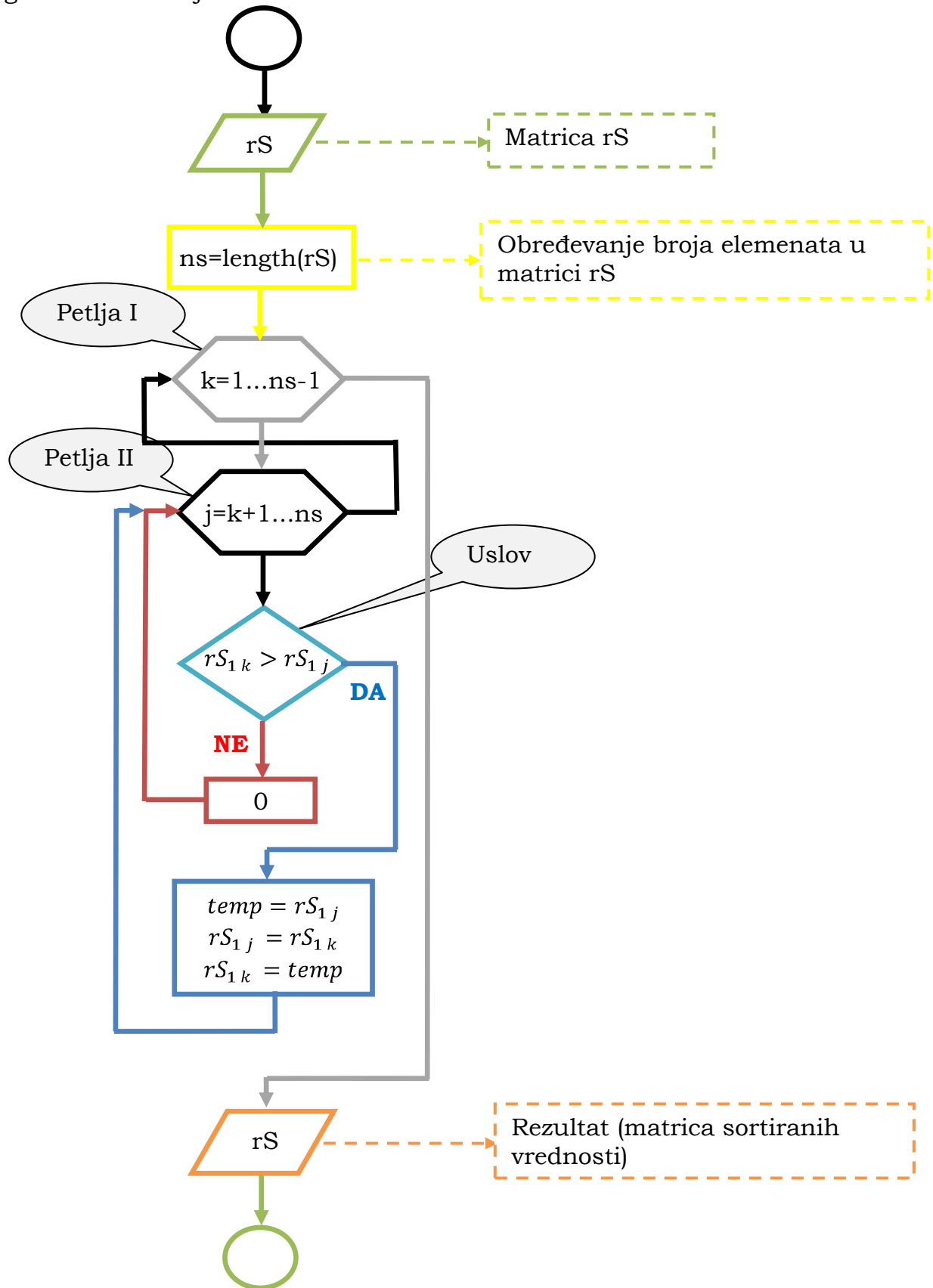
```

A=(5.4 1.2 3.5 10.2 -2.5 4.1)

sort(A^T)=
    (-2.5)
    1.2
    3.5
    4.1
    5.4
    10.2
    
```

Slika 145. For petlja 9

Algoritam rešavanja :



Slika 146. For petlja 10

Sledeći primer ilustruje kako pomoću for petlje možemo nacrtati grafik neke funkcije.

Primer:

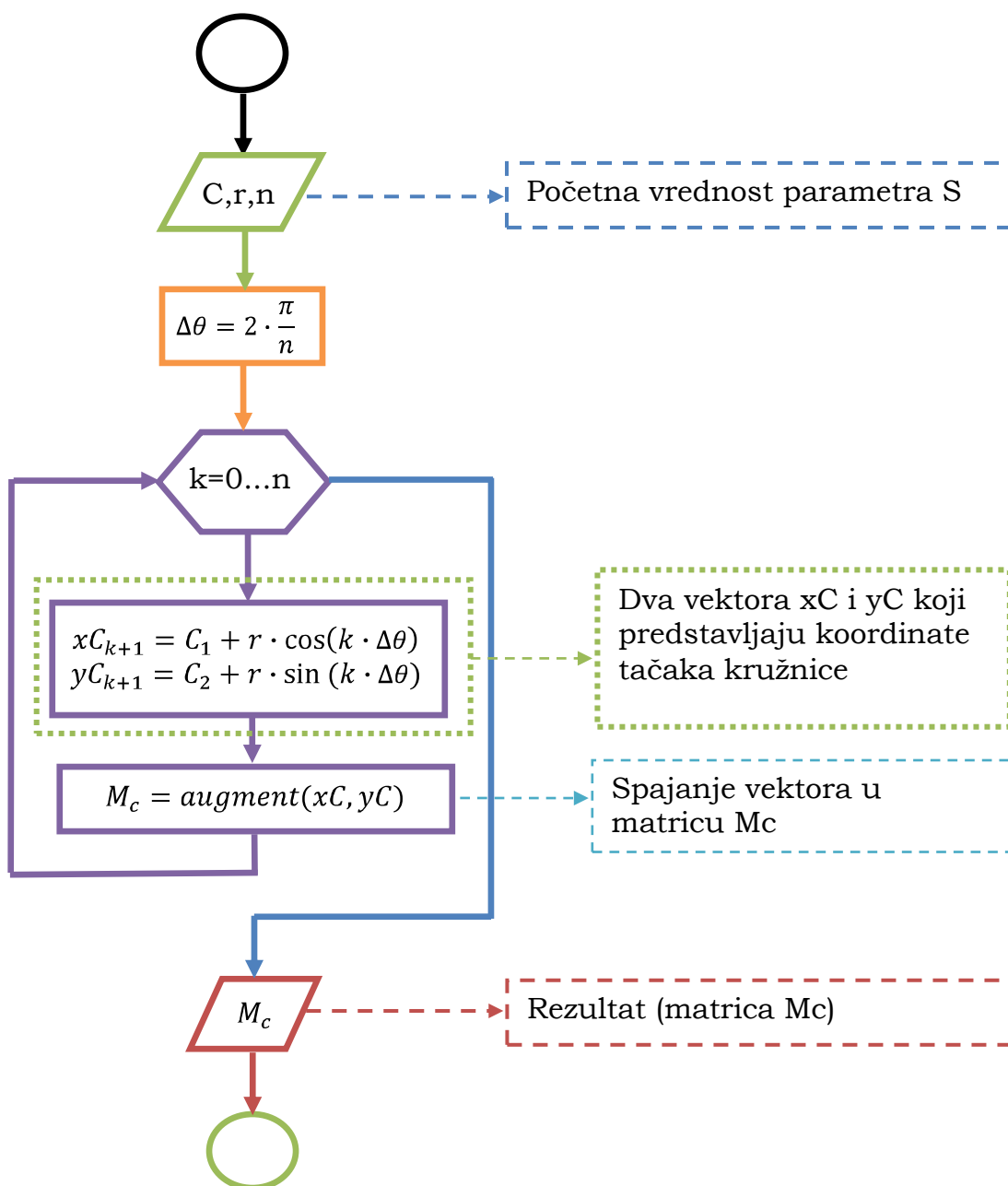
Dat je centar kružnice $C(-3;5)$ i radijus $r=12.5$ m. Nacrtati kružnicu.

Izabrano je da je broj podataka koje ćemo koristiti prilikom crtanja ove funkcije jednak stopedeset.

$n=150$

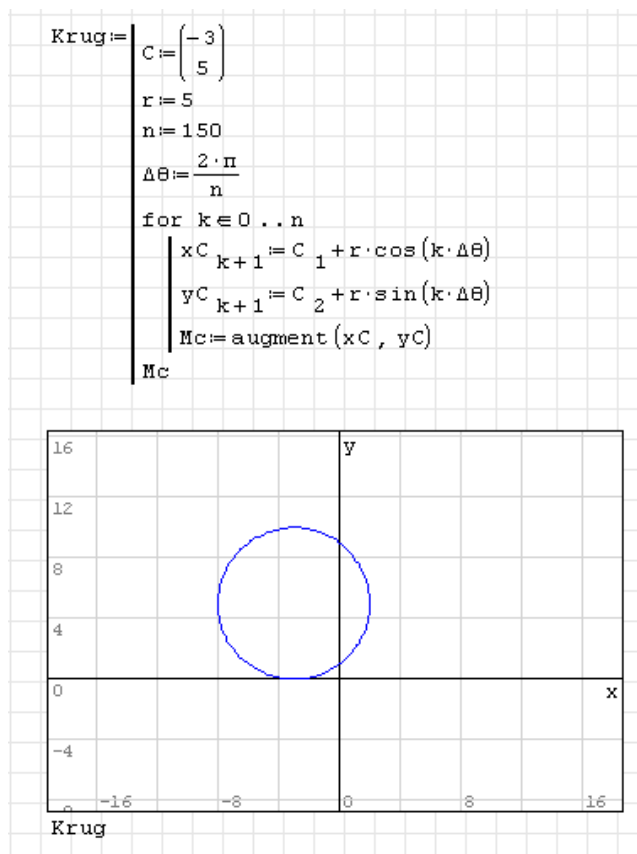
A inkrement je $\Delta\theta = 2 \cdot \frac{\pi}{n} = 2 \cdot \frac{\pi}{150} = 0.0419$

Prvo ćemo prikazati algoritam rešavanja ovog problema.



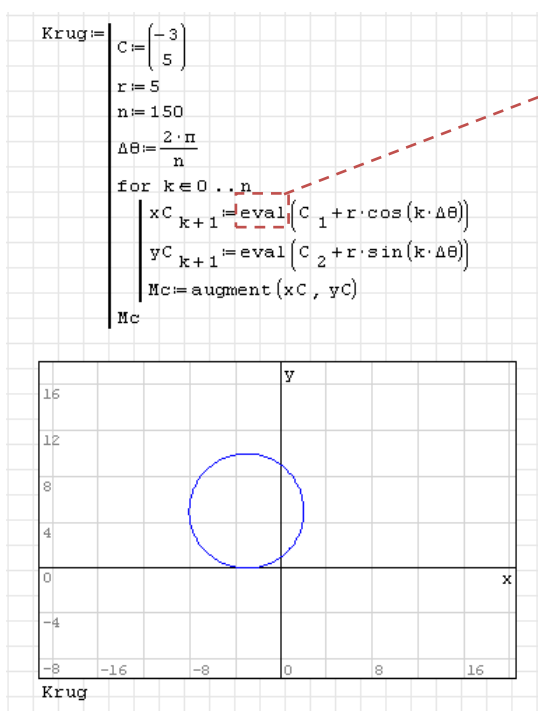
Slika 147. For petlja 11

Rešenje problema primenom programa SMath Studio:



Slika 148. For petlja 12

Za ovaj proračun programu je potrebno 1 min. 19,734 sek. Razlog ovoliko “dugog” proračuna leži u činjenici da SMath prvo izračunava simbolički, a zatim vrednosti tog proračuna pretvara u numeričke vrednosti. Ovo se može promeniti tj. može se ubrzati.



Slika 149. For petlja 13

Funkcija **eval** pretvara analitički izraz u numeričku notaciju.

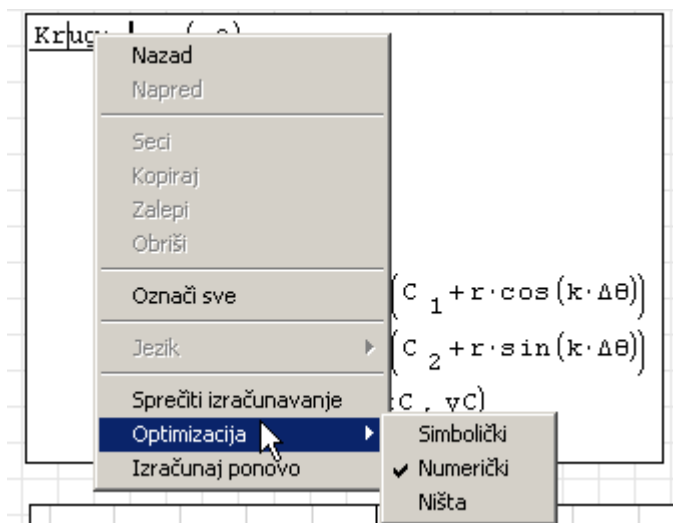
Potrebno je 5,546 sek. da bi se izvršio ovaj program.

Karakteristike računara korišćene u ovom primeru su sledeće:

Pentium(R) 4 CPU 2.00GHz
2.02 GHz, 384 MB of RAM

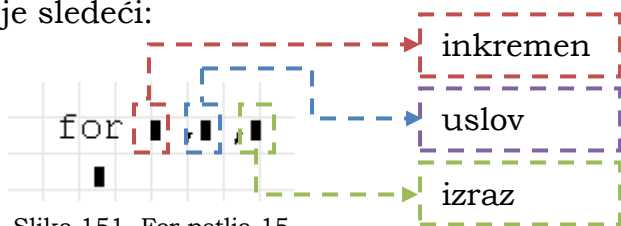
Windows XP Service Pack 3

Program ima mogućnost vršenja optimizacije kako bi se program što brže izvršio. Pristupanje ovom meniju je preko desnog klika i izborom opcije Optimizacija.

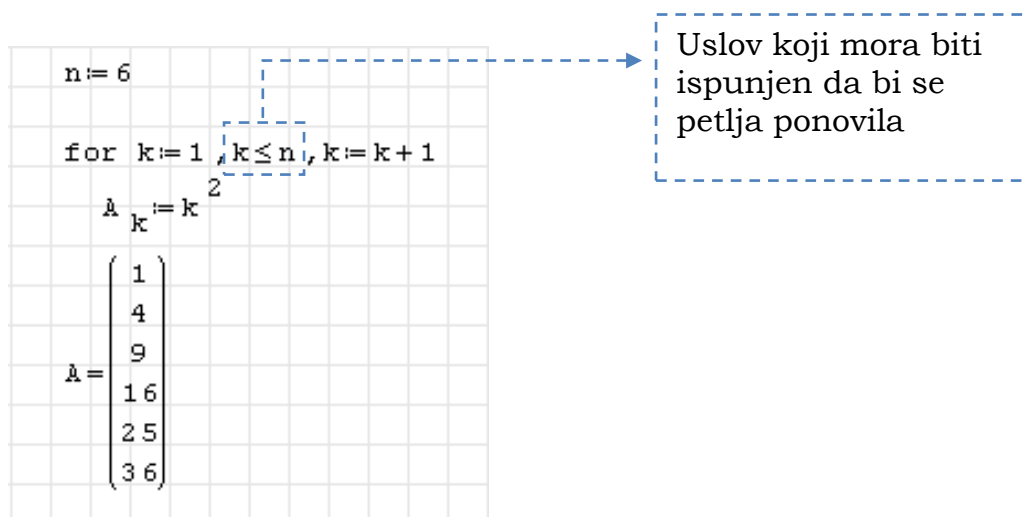


Slika 150. For petlja 14

Ukratko će biti objašnjena funkcija **for(4)** kroz jedan mali primer. Oblik komande je sledeći:



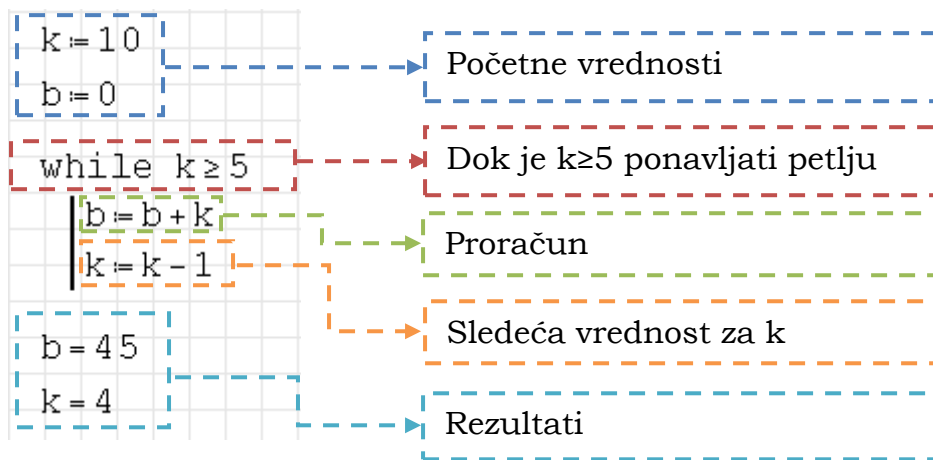
Slika 151. For petlja 15



Slika 152. For petlja 15

WHILE petlja

While petlja se koristi kada unapred ne znamo koliko puta petlja treba da se ponovi. Najlakše ćemo ovu petlju objasniti kroz par primera.



Slika 153. While petlja 1

Prođimo kroz ovu petlju postupno.

I ciklus **$k=10$** , početak petlje $k \geq 5$ ($10 > 5$ uslov zadovoljen) ; $b = 0 + 10 = 10$
 Sledeća vrednost za $k = k - 1 = 10 - 1 = 9$ **$k=9$**

II ciklus **$k=9$** , $k \geq 5$ ($9 \geq 5$ uslov zadovoljen) ; $b = 10 + 9 = 19$; $k = k - 1 = 9 - 1 = 8$ **$k=8$**

III ciklus **$k=8$** , $k \geq 5$ ($8 \geq 5$ uslov zadovoljen) ; $b = 19 + 8 = 27$; $k = k - 1 = 8 - 1 = 7$ **$k=7$**

IV ciklus **$k=7$** , $k \geq 5$ ($7 \geq 5$ uslov zadovoljen) ; $b = 27 + 7 = 34$; $k = k - 1 = 7 - 1 = 6$ **$k=6$**

V ciklus **$k=6$** , $k \geq 5$ ($6 \geq 5$ uslov zadovoljen) ; $b = 34 + 6 = 40$; $k = k - 1 = 6 - 1 = 5$ **$k=5$**

VI ciklus **$k=5$** , $k \geq 5$ ($5 \geq 5$ uslov zadovoljen) ; $b = 40 + 5 = 45$; $k = k - 1 = 5 - 1 = 4$ **$k=4$**

VII ciklus **$k=4$** , $k \geq 5$ ($4 \geq 5$ uslov nije zadovoljen \rightarrow kraj programa)

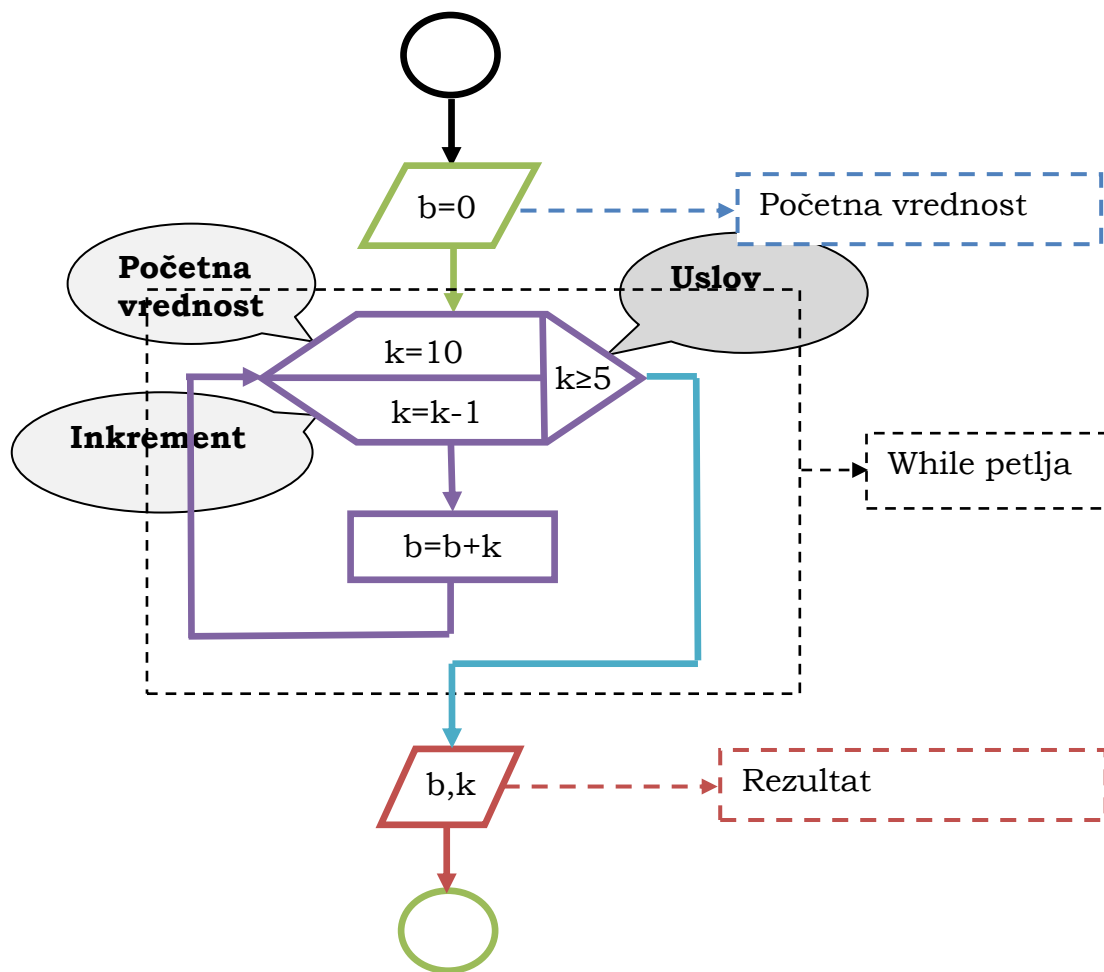
Rezultati:

$k=4$ **$b=45$**

Iz slike 153. uviđamo sledeće:

- Pre same petlje mora se staviti početna vrednost $k=0$
- mora postojati uslov koji mora biti zadovoljen da bi se petlja izvršila
- Posle proračuna mora doći vrednost k posle ciklusa ($k = k - 1$)

Na sledećoj strani prikazan je algoritam rešavanja ovog problema.



Slika 154. While petlja 2

Sada ćemo prikazati rešenje problema

$$\sum_{k=1}^5 \sum_{j=1}^5 (k \cdot j) = 225$$

Ovo smo ranije rešili primenom for petlje.

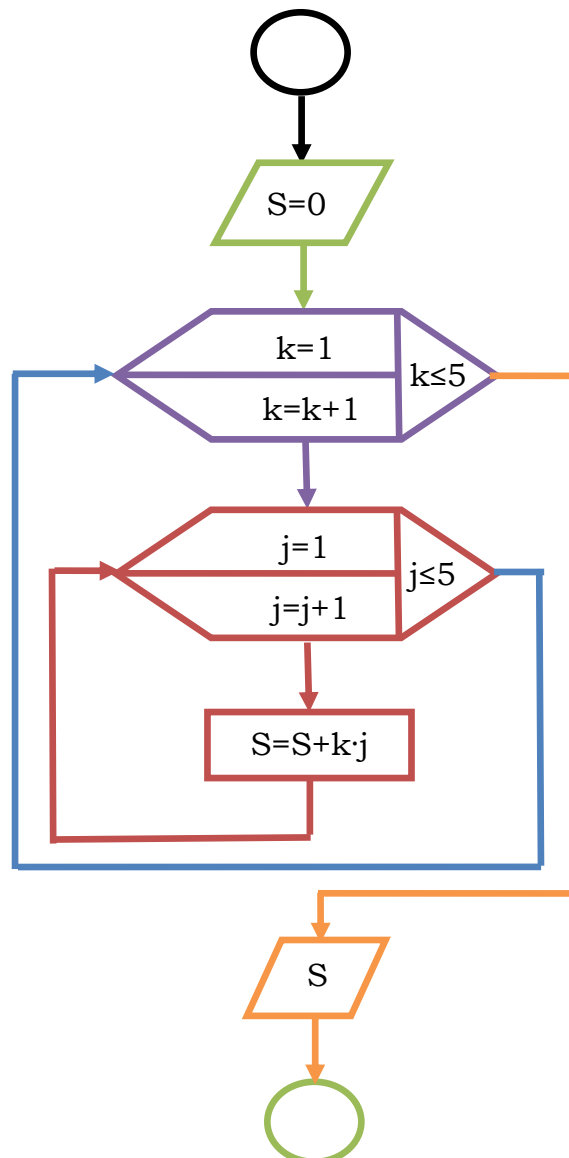
SMath Studio :

```

S0 := S := 0
      k := 1
      while k ≤ 5
          j := 1
          while j ≤ 5
              S := S + k · j
              j := j + 1
          k := k + 1
      S
S0 = 225
    
```

Slika 155. While petlja 3

Algoritam rešavanja.



Slika 156. While petlja 4

Spisak funkcija

Matrice i vektori

- **alg**("1:matrica", "2:broj", "3:broj") - Algebarski dodatak matrici.

$$A_{1\ 2} \left(\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \right) = 6$$

- **augment(...)** - Vraća matricu tako što se argumenti spajaju sleva nadesno. Argumenti su nizovi sa istim brojem kolona, ili skalari i vektori kolone.

$$\text{augment} \left(\begin{pmatrix} 3 \\ 1 \\ -2 \end{pmatrix}, \begin{pmatrix} 5 & -6 \\ 5 & -3 \\ 0 & 1 \end{pmatrix} \right) = \begin{pmatrix} 3 & 5 & -6 \\ 1 & 5 & -3 \\ -2 & 0 & 1 \end{pmatrix}$$

- **col**("1:matrica", "2:broj") - Broj kolone matrice/vektora.

$$\text{col} \left(\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}, 2 \right) = \begin{pmatrix} 2 \\ 5 \end{pmatrix}$$

- **cols**("matrica") - Broj kolona matrice/vektora.

$$\text{cols} \left(\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \right) = 3$$

- **csort**("1:matrica", "2:broj") - Vraća modifikovanu matricu/vektor zamenivanjem mesta vstama, pri čemu se data kolona sortira po rastućem redosledu.

$$\text{csort} \left(\begin{pmatrix} 3 & 0 & -5 \\ 1 & 3 & 0 \\ 0 & 2 & 1 \end{pmatrix}, 1 \right) = \begin{pmatrix} 0 & 2 & 1 \\ 1 & 3 & 0 \\ 3 & 0 & -5 \end{pmatrix}$$

- **det**("matrica") - Determinanta matrice.

$$\begin{vmatrix} 3 & i & -5 \\ 1 & 3 & i \\ i & 2 & 1 \end{vmatrix} = -1 + 7 \cdot i$$

- **diag**("broj") - Vraća dijagonalnu matricu sa elementima vektora na glavnoj dijagonali.

$$\text{diag} \left(\begin{pmatrix} 5 \\ -6 \\ \sqrt{3} \end{pmatrix} \right) \rightarrow \begin{pmatrix} 5 & 0 & 0 \\ 0 & -6 & 0 \\ 0 & 0 & \sqrt{3} \end{pmatrix}$$

- **el**("1:vektor", "2:broj") - Element matrice/vektora.

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix}_2 = 2$$

- **el**("1:matrica", "2:broj", "3:broj") - Vraća element matrice "1:matrica" u redu "2:broj" i koloni "3:broj".

$$\left(\begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \end{array} \right)_{2 \ 2} = 5$$

- **identity**("matrica") - Računa n x n jedinicnu matricu (matrica 0 sa 1 na dijagonali). n mora biti pozitivan ceo broj.

$$\text{identity}(2) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

- **invert**("matrica") - Inverzna vrednost.

$$\text{invert} \left(\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \right) = \begin{pmatrix} -2 & 1 \\ 1.5 & -0.5 \end{pmatrix}$$

- **length**("matrica") - Broj elemenata u vektoru ili matrici.

$$\text{length} \left(\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \right) = 6$$

- **mat**(...) - Matrica.

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

- **matrix**("1:broj", "2:broj") - Vraća nula matricu veličine određene argumentima.

$$\text{matrix}(2, 4) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

- **max**("matrica") - Najveći element matrice ili vektora. Ako je bilo koja vrednost kompleksna, računa $\max(\text{Re}(\dots)) + * \max(\text{Im}(\dots))$.

$$\max \left(\begin{pmatrix} -1 & 2 \\ -3 & 4 \end{pmatrix} \right) = 4$$

- **min**("matrica") - Najmanji element vektora ili matrice. Ako je bilo koja vrednost kompleksna, računa $\min(\text{Re}(\dots)) + * \min(\text{Im}(\dots))$.

$$\min \left(\begin{pmatrix} -1 & 2 \\ -3 & 4 \end{pmatrix} \right) = -3$$

- **minor**("1:matrica", "2:broj", "3:broj") - Minor matrice.

$$M_{1\ 2} \left(\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \right) = -6$$

- **norm1**("matrica") - Vraća L1 normu matrice.

$$\text{norm1} \left(\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \right) = 6$$

- **norme**("matrica") - Vraća Euklidnu normu matrice

$$\text{norme} \left(\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \right) = 5.48$$

- **normi**("matrica") - Vraća ravnomernu normu matrice

$$\text{normi} \left(\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \right) = 7$$

- **rank**("matrica") - Rang matrice.

$$\text{rank} \left(\begin{pmatrix} 4 & 2 & 2 \\ 3 & 2 & 2 \\ 3 & 2 & 2 \end{pmatrix} \right) = 2$$

- **reverse**("matrica") - Obrće redosled redova u matrici, elemenata vektora.

$$\text{reverse} \left(\begin{pmatrix} 7 \\ -1 \\ 2 \end{pmatrix} \right) = \begin{pmatrix} 2 \\ -1 \\ 7 \end{pmatrix}$$

- **row**("1:matrica", "2:broj") - Broj reda matrice/vektora.

$$\text{row} \left(\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}, 1 \right) = (1\ 2\ 3)$$

- **rows**("matrica") - Broj redova matrice/vektora.

$$\text{rows} \left(\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \right) = 2$$

- **rsort**("1:matrica", "2:broj") - Vraća modifikovanu matricu/vektor zamenivanjem mesta kolonama, pri čemu se data vrsta sortira po rastućem redosledu.

$$\text{rsort} \left(\begin{pmatrix} 3 & 0 & -5 \\ 1 & 3 & 0 \\ 0 & 2 & 1 \end{pmatrix}, 1 \right) = \begin{pmatrix} -5 & 0 & 3 \\ 0 & 3 & 1 \\ 1 & 2 & 0 \end{pmatrix}$$

- **sort**("matrica") - Vraća vektor sa vrednostima sortiranim u rastućem redosledu.

$$\text{sort} \left(\begin{pmatrix} 7 \\ -1 \\ 2 \end{pmatrix} \right) = \begin{pmatrix} -1 \\ 2 \\ 7 \end{pmatrix}$$

- **stack**(...) - Vraća vektor formiran od argumenata od vrha na dole. Argumenti su nizovi sa istim brojem kolona, ili su skalari i vektori kolone.

$$\text{stack} \left((3 \ 1 \ -2), \begin{pmatrix} 5 & -6 & 5 \\ -3 & 0 & 1 \end{pmatrix} \right) = \begin{pmatrix} 3 & 1 & -2 \\ 5 & -6 & 5 \\ -3 & 0 & 1 \end{pmatrix}$$

- **submatrix**("1:matrica", "2:argument", "3:argument", "4:argument", "5:argument") - Podmatrica koja se sastoji od elemenata u redovima i_red do j_red i kolonama i_kol do j_kol.

$$\text{submatrix} \left(\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}, 2, 3, 1, 3 \right) = \begin{pmatrix} 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

- **tr**("matrica") - Trag matrice.

$$\text{tr} \left(\begin{pmatrix} 3 & 2 \\ 3 & 1 \end{pmatrix} \right) = 4$$

- **transpose**("matrica") - Transponovana matrica.

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}^T = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}$$

- **vminor**("1:matrica", "2:broj", "3:broj") - Podmatrica date matrice izuzev odabranog reda i kolone.

$$M_{2 \ 2} \left(\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \right) = \begin{pmatrix} 1 & 3 \\ 7 & 9 \end{pmatrix}$$

Kompleksni brojevi

- **arg**("kompleksanBroj") - Ugao od realne ose do kompleksnog broja.

$$\text{arg}(3 + 3 \cdot i) = 0.79$$

- **Im**("kompleksanBroj") - Imaginarni deo kompleksnog broja.

$$\text{Im}(5 + i \cdot 2) = 2$$

- **pol2xy**("1:argument", "2:argument") - Pretvara polarne koordinate tačke u 2D prostoru u pravougle koordinate.

$$\text{pol2xy}(5, 0.9273) = \begin{cases} 3 - 3.11 \cdot 10^{-15} \cdot i \\ 4 + 2.43 \cdot 10^{-15} \cdot i \end{cases}$$

- **Re**("kompleksanBroj") - Realni deo kompleksnog broja.

$$\text{Re}(5 + i \cdot 2) = 5$$

- **xy2pol**("1:argument", "2:argument") - Pretvara pravougle koordinate tačke u 2D prostoru u polarne koordinate.

$$\text{xy2pol}(3, 4) = \begin{cases} 5 \\ 0.93 \end{cases}$$

Trigonometrija

- **arccos**("kompleksanBroj") - Inverzni kosinus.

$$\text{arccos}(1) = 0$$

- **arccosec**("kompleksanBroj") - Inverzni kosekans.

$$\text{arccosec}(\infty) = 0$$

- **arcctg**("kompleksanBroj") - Inverzni kotangens.

$$\text{arcctg}(1) \rightarrow \frac{\pi}{4}$$

- **arcsec**("kompleksanBroj") - Inverzni sekans.

$$\text{arcsec}(1) = 0$$

- **arcsin**("kompleksanBroj") - Inverzni sinus.

$$\text{arcsin}(0) = 0$$

- **arctg**("kompleksanBroj") - Inverzni tangens.

$$\text{arctg}(\infty) \rightarrow \frac{\pi}{2}$$

- **cos**("kompleksanBroj") - Kosinus.

$$\cos(0) = 1$$

- **cosec**("kompleksanBroj") - Kosekans.

$$\text{cosec}(1) = 1.19$$

- **ctg**("kompleksanBroj") - Kotangens.

$$\text{ctg}(1) = 0.64$$

- **sec**("kompleksanBroj") - Sekans.

$$\sec(1) = 1.85$$

- **sin**("kompleksanBroj") - Sinus.

$$\sin(0) = 0$$

- **tg**("kompleksanBroj") - Tangens.

$$\text{tg}(0) = 0$$

Hiperbolični

- **arch**("kompleksanBroj") - Inverzni kosinus hiperbolični.

$$\text{arch}(0) = 1.57 \cdot i$$

- **arth**("kompleksanBroj") - Inverzni kotangens hiperbolični.

$$\text{arth}(2) = 0.55$$

- **arsh**("kompleksanBroj") - Inverzni sinus hiperbolični.

$$\text{arsh}(0)=0$$

- **arth**("kompleksanBroj") - Inverzni tangens hiperbolični.

$$\text{arth}(0)=0$$

- **ch**("kompleksanBroj") - Kosinus hiperbolični.

$$\text{ch}(1)=1.54$$

- **cosech**("kompleksanBroj") - Kosekans hiperbolični.

$$\text{cosech}(1)=0.85$$

- **cth**("kompleksanBroj") - Kotangens hiperbolični.

$$\text{cth}(1)=1.31$$

- **sech**("kompleksanBroj") - Sekans hiperbolični.

$$\text{sech}(1)=0.65$$

- **sh**("kompleksanBroj") - Sinus hiperbolični.

$$\text{sh}(1)=1.18$$

- **th**("kompleksanBroj") - Tangens hiperbolični.

$$\text{th}(1)=0.76$$

Programiranje

- **for**("1:inkrement", "2:izraz", "3:izraz") - For petlja. Funkcija kontrolisanih iteracija. U ciklusu se ponavlja "3:izraz" dok "1:inkrement" uzima vrednosti iz "2:izraz". U "3:izraz" može biti zadato bilo koji broj izraza.

```
for arg1 ∈ arg2
  arg3
```

- **for**("1:inkrement", "2:uslov", "3:izraz", "4:izraz") - For petlja.

```
for arg1 , arg2 , arg3
  arg4
```

- **if**("1:uslov", "2:true", "3:false") - Računa 'tacno'-tvrdnju ako je 'uslov' ispunjen (ne-nula), a inače 'pogresno'-tvdnju.

```
if arg1
  arg2
else
  arg3
```

- **line**(...) - Služi za pisanje linija potprograma.
- **while**("1:uslov", "2:izraz") - Iteraciona funkcija. Ciklus prikazuje 'objekat' kada je 'uslov' istinit. Važno: u 'objekat' može se postaviti bilo koja vrednost ili izraz pomoću funkcije line(...).

```
while arg1
  arg2
```

Stringovi

- **importData**("1:imeFajla", "2:separator", "3:separator", "4:separator", "5:broj", "6:broj", "7:broj", "8:broj", "9:broj") - Vraća matricu iz datog fajla. Funkcija dozvoljava 1-9 argumenata. Podrazumevani parametri se specificiraju nulom (0) kao argumentom (osim 'imefajla').

```
importData(arg1 , arg2 , arg3 , arg4 , arg5 , 0 , 0 , 0 , 0 , 0)
```

- **IsString**("izraz") - Vraća 1 ako je dati izraz tekstualni. U suprotnom vraća 0.

```
IsString(arg1)
```

- **num2str**("izraz") - Konvertuje matematički izraz u tekstualni.

```
num2str(arg1)
```

- **str2num**("string") - Konvertuje tekstualni izraz u matematički.

```
str2num(arg1)
```

- **strlen**("string") - Vraća broj datih karaktera u tekstualnom izrazu.

```
strlen(arg1)
```

- **strrep**("1:string", "2:string", "3:string") - Zamenjuje sve pojave teksta stariSubstr u originalStr sa noviSubstr.

```
strrep(arg1 , arg2 , arg3)
```

- **substr**("1:string", "2:broj") - Vraća deo teksta počev od date pozicije. Drugi argument daje početak a treći (ako je zadat) dužinu rezultujućeg teksta.

```
substr(arg1 , arg2)
```

- **substr**("1:string", "2:broj", "3:broj") - Vraća podstring od "1:string", gde je "2:broj" početna pozicija podstringa a "3:broj" je dužina podstringa.

```
substr(arg1 , arg2 , arg3)
```

Fajlovi

- **dfile**("imeFajla") - Briše fajl iz sistema, ako postoji. Vraća '1' po uspešnom brisanju, '0' u suprotnom.

```
dfile(arg1)
```

- **importData**("imeFajla") - Vraća matricu sa elementima datim u fajlu koristeći podrazumevane parametre.

```
importData(arg1)
```


- **importData**("1:imeFajla", "2:separator", "3:separator", "4:separator", "5:broj", "6:broj", "7:broj", "8:broj", "9:broj") - Vraća matricu iz datog fajla. Funkcija dozvoljava 1-9 argumenata. Podrazumevani parametri se specificiraju nulom (0) kao argumentom (osim 'imefajla').

```
importData(arg1 , arg2 , arg3 , arg4 , arg5 , ...)
```

- **rfile**("imeFajla") - Učitava matematički izraz iz fajla, ako fajl sa tim imenom postoji. Često vraća simbolički rezultat

```
rfile(arg1)
```

- **wfile**("1:izraz", "2:imeFajla") - Upis matematičkog izraza u fajl. Ako fajl sa datim imenom postoji, prethodni sadržaj fajla će se poništiti. Funkcija vraća '1' ako operacija upisa uspe, '0' u suprotnom.

```
wfile(arg1 , arg2)
```