



**KAUNO TECHNOLOGIJOS UNIVERSITETAS  
ELEKTROS IR ELEKTRONIKOS FAKULTETAS**

**Rytis Bendaravičius**

**PROGRAMINĖS APLINKOS NERAIŠKIŲ AIBIŲ LOGIKOS  
SISTEMŲ REALIZAVIMUI SUKŪRIMAS**

Baigiamasis magistro projektas

**Vadovas**

prof. dr. Rimvydas Simutis

**KAUNAS, 2016**

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**ELEKTROS IR ELEKTRONIKOS FAKULTETAS**  
**AUTOMATIKOS KATEDRA**

**PROGRAMINĖS APLINKOS NERAIŠKIŲ AIBIŲ LOGIKOS  
SISTEMŲ REALIZAVIMUI SUKŪRIMAS**

Baigiamasis magistro projektas  
M3066L21 Valdymo technologijos (621H66001)

**Vadovas**

(parašas) prof. dr. Rimvydas Simutis  
(data)

**Recenzentas**

(parašas)  
(data)

**Projektą atliko**

(parašas) Rytis Bendaravičius  
(data)

**KAUNAS, 2016**



KAUNO TECHNOLOGIJOS UNIVERSITETAS  
Elektros ir elektronikos fakultetas

(Fakultetas)

Rytis Bendaravičius

(Studento vardas, pavardė)

M3066L21 Valdymo technologijos (621H66001)

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Programinė aplinka neraiškių aibių logikos sistemų realizavimui“

### AKADEMINIO SAŽININGUMO DEKLARACIJA

20 \_\_\_\_ m. \_\_\_\_ d.

Kaunas

Patvirtinu, kad mano **Rytis Bendaravičius** baigiamasis projektas tema „Programinės aplinkos neraiškių aibių logikos sistemų realizavimui sukūrimas“ yra parašytas visiškai savarankiškai, o visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

\_\_\_\_\_  
(vardą ir pavardę įrašyti ranka)

\_\_\_\_\_  
(parašas)

Bendaravičius, R. Programinės aplinkos neraiškių aibių logikos sistemų realizavimui sukūrimas. Valdymo sistemų magistro baigiamasis projektas / vadovas prof. dr. Rimvydas Simutis; Kauno technologijos universitetas, elektros ir elektronikos fakultetas, automatikos katedra.

Kaunas, 2015. 120 psl.

## SANTRAUKA

Darbo tikslas yra sukurti programinį paketą, kuriuo būtų galima kurti neraiškių aibių sistemas ir realizuoti neraiškių aibių sistemų hierarchines struktūras pasitelkus netradicines neraiškių aibių sistemos įėjimų/išėjimų funkcines priklausomybes. Darbe apžvelgiama: neraiškių aibių logika, pagrindiniai programiniai paketai skirti kurti neraiškių aibių sistemas ar atlikti neraiškių aibių logikos skaičiavimus, sukurtas programinis paketas, jo struktūra, programavimo kalba, programavimo terpė ir trumpa naudojimo instrukcija bei du taikymo pavyzdžiai. Neraiškių aibių logika leidžia patogiai formalizuoti procesų eksperimentų žinias ir sprendimus, tačiau šiuo metu programiniuose paketuose neraiškumo eliminavimui dažniausiai naudojamas apjungtų taisyklių išėjimo kintamojo priklausomybės svorio centro nustatymo metodas, kuris nevertina išėjimo kintamojo apibrėžtumo. Analizė parodė, kad egzistuojantys neraiškių aibių logikos programiniai paketai turi savo pranašumų ir trūkumų. Vieni nėra autonomiški, kiti brangūs, dar kiti reikalauja daugiau priežiūros. Tačiau dauguma analizuotų paketų defuzifikavimo procedūrai naudoja apjungtų taisyklių svorio centro principą. Sukurto programinio paketo įėjimų/išėjimų funkcinės priklausomybės leidžia įvertinti sistemos narystės funkcijų neraiškumus, kadangi defuzifikavimo procedūroje narystės funkcijos traktuojamos, kaip tikimybinių skirstinių aproksimacija. Paketo programavimo terpė yra Microsoft Visual Studio 2015, o programavimo kalba - Visual C#, kuri palaiko .NET struktūrą. Todėl sukurtas programinis paketas FuzzyThis gali būti lengvai integruotas į programinę įrangą veikiančią su .NET struktūra. FuzzyThis turi visas reikalingas funkcijas kurti neraiškių aibių sistemas ir galimybę kurti hierarchinės struktūros neraiškių aibių sistemas. Pasiūlyta neraiškių aibių sistema ypač tinka tokiems atvejams, kada įvairių taisyklių išėjimo kintamųjų priklausomybės funkcijos turi skirtingus pločius (t.y. labai skiriasi taisyklių taikymo neapibrėžtumas). Kada priklausomybės funkcijų pločiai yra panašūs, gaunami sistemos išėjimai yra panašūs, kaip ir taikant tradicines neraiškių aibių sistemas. Pasiūlytos įrangos testavimas parodė, kad FT aplinkoje sudaryta hierarchinė neraiškių aibių sistema tinkamai identifikuoja S&P 500 indekso kainos kitimo mažėjimo ir didėjimo fraktalus. Programos testai taip pat parodė, kad FT aplinkoje apskaičiuoti hierarchinės ekspertinės investuotojo rizikos sistemos rezultatai yra panašūs su kitų autorių gautais rezultatais, todėl pasiūlytą sistemą, kaip autonominę programinę įrangą, galima sėkmingai taikyti ir kituose hierarchinėse neraiškių aibių sistemose.

*Reikšminiai žodžiai: Neraiškių aibių logika, programinė įranga, Fraktalas, Visual C#, MATLAB.*

Bendaravičius, Rytis. Development of Software Environment for Implementation of Fuzzy Logic Systems. Final project of *Control systems master's degree* / supervisor prof. dr. Rimvydas Simutis; Kaunas University of Technology, Faculty of Electrical and Electronics Engineering, department of automation.

Kaunas, 2016. 120 psl.

## SUMMARY

The goal of this Master thesis is to create software, which could be used for creating Fuzzy systems and implementing fuzzy hierarchical structures using non-traditional fuzzy input/output functional dependencies. This Master thesis provides an overview of fuzzy logic, main software for implementing fuzzy systems or performing fuzzy calculations, created software, its structure, programming language, programming environment, short user manual and two examples of application. Fuzzy logic allows to comfortably formalize expert knowledge and decisions, however currently, in existing fuzzy logic software, uncertainty is eliminated using combined rules output variable weight center determination method, which do not estimate output variable certainty. Analysis showed, that existing fuzzy logic software have its advantages and disadvantages. Some is not autonomous, some are expensive, some requires more maintenance. However majority of analyzed software, for defuzzification procedure, use combined rules weight center principle. Created software input/output functional dependences allow to measure system's membership function uncertainties, because in defuzzification procedure membership functions are treated as probability distribution approximations. Software is created in Microsoft Visual Studio 2015 programming environment and chosen programming language is Visual C#, which supports .NET framework. So created software FuzzyThis can be easily integrated in software, which runs on .NET framework. FT has all necessary functions to create fuzzy systems and option to create hierarchical structure fuzzy logic systems. Offered fuzzy logic system especially useful in such cases, when different rule's output variable membership function has different widths (eg. Rule application uncertainty is very different). When membership function width is similar, systems output is similar like using traditional fuzzy logic systems. Offered software testing showed, that hierarchical fuzzy logic system, implemented in FT environment, properly identify S&P 500 price index fluctuation downward and upward fractals. Software tests also showed, that hierarchical expert investor risk system, implemented in FT environment, calculated results is similar with other authors results, so offered system, as autonomous software, can be successfully used in others hierarchical fuzzy logic systems.

*Keywords: Fuzzy logic, software, fractal, Visual C#, MATLAB*

## Turinys

Įvadas .....	1
1. Neraiškių aibių logikos analizė .....	1
1.1 Neraiški logika.....	2
1.2 Neraiškios logikos rinkiniai ir neraiškios logikos nariai .....	4
1.3 Neraiškių žinių modelis .....	8
2. Programinės įrangos apžvalga .....	11
2.1 MATLAB Fuzzy Logic Toolbox.....	11
2.2 FuzzyTECH.....	11
2.3 CubiCalc .....	12
2.4 FuzzyCLIPS.....	13
3. Sukurtas programinis paketas FuzzyThis (FT) .....	14
3.1 FT neraiškių aibių sistemos įėjimų /išėjimų funkcinės priklausomybės .....	14
3.2 FT programavimo terpė ir kalba .....	15
3.3 FT vartotojo sąsaja.....	17
3.4 MATLAB Fuzzy Logic Toolbox ir FuzzyThis rezultatų palyginimas.....	26
4. Taikymo pavyzdžiai.....	31
4.1 Akcijų kainos augimo/kritimo nustatymas naudojant neraiškių aibių sistemą realizuotą FuzzyThis (FT) aplinkoje.....	31
4.2 Ekspertinės investuotojo tolerancijos rizikai įvertinimo sistemos realizavimas FuzzyThis (FT) aplinkoje.....	37
5. Išvados .....	51
6. Literatūra.....	52
7. Priedai .....	53

## Ivadas

Neraiški logika yra naudinga tada, kuomet valdymo sistemai reikia priimti sprendimus remiantis lingvistinėmis vertėmis. Beveik kiekviena valdymo sistema gali būti pakeista neraiškia logika paremta valdymo sistema. Šis teiginys gali turėti trūkumų daugelyje atvejų, tačiau neraiški logika palengvina sudėtingų procesų ar sistemų valdymą. Tarkime, jei paprasta uždara sistema ar PID valdiklis dirba gerai – tai neraiškaus valdiklio nereikia, tačiau daugeliu atvejų pritaikyti PID valdiklį arba planuoti valdymo sistemą sudėtingai sistemai yra per daug sudėtinga, tokiu atveju neraiški logika turi didžiausią potencialą. Modeliuojant sistemas šis principas taip pat tinka, tačiau šiuo atveju susiduriama su programinės įrangos ir metodikos įvairove. Dažniausiai neraiškių aibių sistemų kūrimui naudojamas paketas yra MATLAB Fuzzy Logic Toolbox. Dar naudojami paketai: fuzzyTECH, CubiCalc, FIDE, FlexTool, FLINT, FuzzyCLIPS, Fuzzy Query, TILShell. Kadangi MATLAB programinis paketas yra labiausiai paplitęs, tobulinimą ir rezultatus lyginsiu su gautais šiuo paketu ir taipogi palyginimui apžvelgsiu dar kelis alternatyvius paketus: fuzzyTECH, CubiCalc, FuzzyCLIPS. Egzistuojančių paketų trūkumai: pirma problema yra autonomiškumas, pavyzdžiui MATLAB Fuzzy Logic Toolbox paketas yra tik dalelė viso paketo ir tai gali būti nepriimtina sprendžiant tam tikras neraiškių aibių sistemos problemas. Antra problema - hierarchija, kai susiduriama su daugybe įėjimų, jiems aprašyti taisyklių bazę yra labai sudėtinga, nes taisyklių yra labai daug, ir reikėtų sistemą išskirstyti į tam tikras posistemas su savo įėjimais ir taisyklėmis. Trečia problema su kuria gali susidurti vartotojai yra neapibrėžtumas, kuomet labiau apibrėžtos reikšmės yra neįvertinamos. Taigi norint išspręsti šias kilusias problemas programuosiu neraiškių aibių programinį paketą ir panaudosiu neraiškių aibių modelių sudarymui.

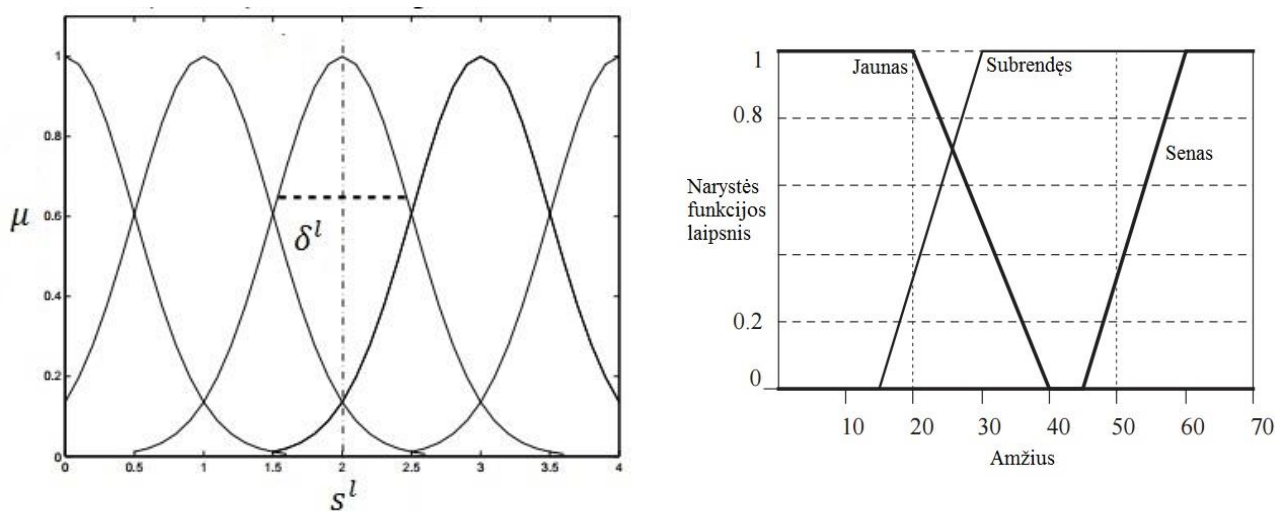
# 1. Neraiškių aibių logikos analizė

## 1.1 Neraiški logika

Teiginį truputį perkrautas galios transformatorius gali atlaikyti šį krūvį kurį laiką patyręs inžinierius supras labai lengvai, kadangi jis turi žinių pagrindus ir yra praręs, kad teiginiai būtų pateikiami tokia forma, tačiau kompiuteris tokio teiginio nesupras. Taigi kaip suteikti panašų teiginio supratimą kompiuteriui? Kaip padaryti, kad kompiuteris galėtų suprasti neraiškius ir dviprasmiškus teiginius? Šias problemas išsprendžia neraiški logika.

Klasikinėje rinkinių teorijoje elementas gali pilnai praklaudyti rinkiniui arba nepriklaudyti. Neraiškios logikos teorijoje elementas gali priklausyti rinkiniui tam tikru laipsniu [0-1], tokiu atveju teiginį minėtą pradžioje galima aprašyti neraiškios logikos narystės funkcijomis ir kompiuteris turės panašų supratimą, kaip ir patyręs inžinierius.

Pavyzdžiui sąvokos „senas“ ir „jaunas“ yra netikslios, bet lengvai interpretuojamos konkrečiame kontekste. Tam tikrose aplikacijose, pavyzdžiui, ekspertinėse sistemose, yra būtina panaudoti metodus galinčius spręsti tokio tipo išraiškas, kad kompiuteris, naudodamasis Būlio logika, galėtų apdoroti informaciją. Būtent tokių tikslų neraiškių rinkinių teorija ir neraiški logika bando pasiekti.



**1.1 pav.** Gauso narystės funkcijos pavyzdys (kairėje) ir „Jaunas“, „Senas“ trapecijos narystės funkcijos pavyzdys (dešinėje)

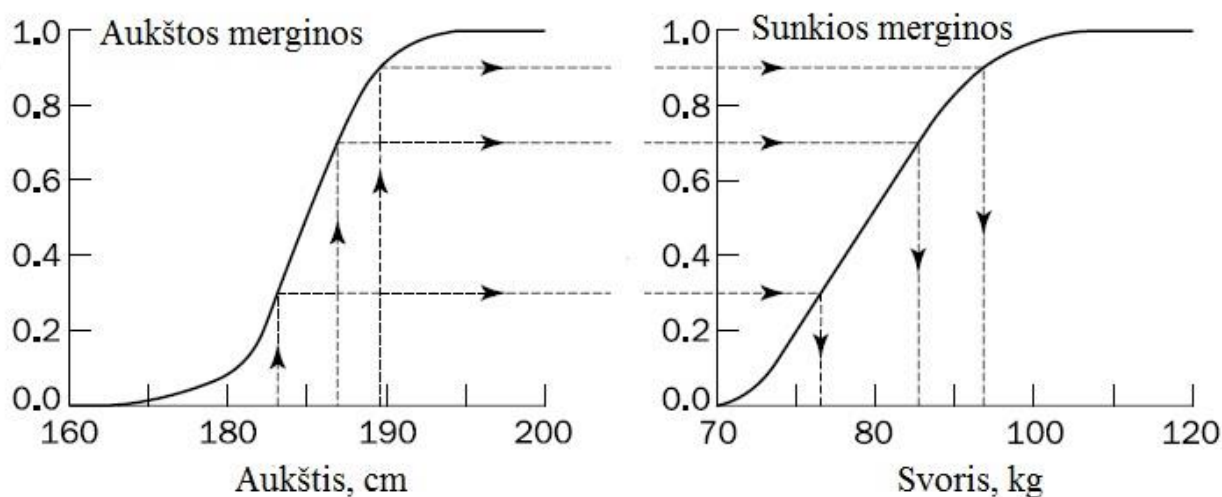
Paveikslėlyje (1.1 pav.) dešinėje yra pavaizduota trys narystės funkcijos intervale nuo nulio iki septyniasdešimt metų. Šios trys funkcijos aprašo narystės laipsnį kiekvienam duotam amžiui rinkiniuose: „jaunas“, „subrendęs“ ir „senas“. Tokiu atveju, jei kas nors yra dvidešimties metų, jo



narystės laipsnis „jaunas“ rinkinyje bus 1.0, „subrendęs“ rinkinyje - 0.35, ir „senas“ rinkinyje – 0.0. Analogiškai. penkiasdešimties metų žmogui narystė laipsniai atrodytų taip: 0.0, 1.0, 0.3.

Neraiškių rinkinių teorijos šaknys yra lingvistiniai kintamieji. Lingvistinis kintamasis yra neraiškus kintamasis. Pavyzdžiui, teiginys „Brigita yra žema“ teigia, jog lingvistiniam kintamajam „Brigita“ priskiriama lingvistinė vertė „žema“. Neraiškiose sistemose, lingvistiniai kintamieji yra panaudojami neraiškiose taisyklėse. Pvz. JEI vėjas yra stiprus TAI skristi lėktuvu negalima.

Klasikinėje taisyklėmis paremtoms sistemoms, jei taisyklė priekis yra tiesa, tai rezultatas taip pat yra tiesa. Neraiškiose sistemose, kur priekis yra neraiškus teiginys, visos taisyklės teisingos tik tam tikru laipsniu, arba dalinai teisingos. Jei priekis yra teisingas tam tikru narystės funkcijos laipsniu, tai rezultatas yra taip pat teisingas tuo pačiu laipsniu (pvz. 3.Pav.).



**1.2 pav.** Taisyklė: JEI aukštis didelis TAI svoris didelis

Norint gauti tikrą išėjimo vertę, neraiški ekspertinė sistema pirma užpildo visus išėjimo neraiškius rinkinius į vieną išėjimo neraiškų rinkinį, ir tada jam atlieka defuzifikavimo procedūrą ir gauna skaičių.

Kai 1965 metais Zadeh pasiūlė neraiškios logikos teoriją, neraiški logika tapo viena iš patraukliausių intelektualaus valdymo sistemų tobulinimo metodikų. Trumpai tariant, žmogaus sprendimai ar valdymo veiksmai yra priimami remiantis netikslia ir arba nepilna sistemos aplinkos informacija, žiniomis ir apytiksliais samprotavimu, kuris skiriasi nuo matematinio modeliu paremtos modernios valdymo technologijos. Elementarus pavyzdys būtų šalto – šilto vandens maišymo temperatūros valdymo sistema. Sumaišyto vandens temperatūra gali būti valdoma, manipuliuojant šalto ar karšto vandens srautą bei jo temperatūrą. Tradiciškai, matematinio modeliu paremta valdymo sistema gali būti suprojektuota grįžtamojo arba tiesioginio ryšio valdymo principu, kuris gali būti aprašytas aiškiu matematinio algoritmu. Tačiau, sistema taip pat gali būti valdoma neraiškios logikos valdikliu,

kuris susideda iš neraiškios logikos, žmonių kokybinėmis žiniomis paremtų, taisyklių. Šiai sistemai pritaikytos neraiškios logikos valdymo pavyzdinės taisyklės atrodytų taip:

FR1: JEI sumaišyto vandens temperatūra yra truputį aukšta, TAI šalto vandens srautas turėtų būti truputį padidintas.

FR2: JEI karšto vandens temperatūra yra labai maža, TAI šalto vandens srautas labai sumažinamas.

Iš valdymo struktūros pusės, galima lengvai suprasti, kad taisyklė FR1 ir taisyklė FR2 atlieka atitinkamą grįžtamo ir tiesioginio ryšio valdymą. Abu sąlyga ir išvada neraiškios logikos taisyklėse yra neraiškios logikos kintamieji, aprašyti nustatytu neraiškios logikos rinkiniu, pvz.: labai žemas, žemas, vidutinis, aukštas, labai aukštas. Papildomai neraiškios logikos taisyklių tiesa arba vadinama „neraiškios logikos pasiūlymas“ gali būti pavaizduotas kaip neraiškios logikos sąlygų įvairovė: „galima“, „beveik neįmanoma“, „galima“ ir t.t.

Apibendrinant galima teigti, kad skaitinis sprendimas yra reikalaujamas sistemų funkciniam įvertinimui ir apskaičiavimui, kaip procesų modeliavimas, įvertinimas, valdymas ir optimizavimas. Taigi, struktūrinės žinios, pvz. neraiškios logikos „JEI – TAI“ taisyklės, nelabai išsprendžia tokias skaitines problemas, tačiau neraiškios logikos teorija ir metodika suteikia efektyvią ir realią techniką koduoti lingvistinius duomenis su netikslią informacija ir žiniomis į skaitinę struktūrą. Todėl, skaitinis sprendimas gali būti rastas naudojantis sistemos modeliu paremtu neraiškia logika.

## 1.2 Neraiškios logikos rinkiniai ir neraiškios logikos nariai

Paprastuose rinkiniuose ir logikos teorijoje, tik dvi reikšmės, pvz. „tiesa ir netiesa“ arba „1 ir 0“, nusako įvykį arba faktą. Pavyzdžiui, kokybinė valdymo sistema, apibūdinanti pagrindinę paprasto rinkinio ir logikos teorijos sąvoką: Baigtinio produkto kokybė,  $x$ , yra padalinta tik į dvi kategorijas („tinkami“ ir „netinkami“). Vertinant ne-neraiškios logikos, kokybės kintamojo erdvės  $X$ , poaibį  $A$  tinkamiems baigtiniams produktams, kurie turi aiškias arba tikslias ribas:

$$A = \{x | x_{min} \leq x \leq x_{max}\} \quad (1.1)$$

Tai produkto kokybė gali būti paprastai identifikuota narystės funkcija:

$$m_A(x) = \begin{cases} 1 & x_{min} \leq x \leq x_{max} \\ 0 & \text{visa kita} \end{cases} \quad (1.2)$$

$m_A(x)$  žymi, kad  $x$  arba yra  $A$  sudėtyje arba nėra. Galiausiai, kadangi narys  $m_A$  pažymi visus realius kokybės skaičius  $x$  į du skaičius  $\{0,1\}$ , paprasti rinkiniai atitinka dviejų verčių logiką. Papildomai dviejų verčių logika, kai kuriais atvejais gali būti aprašyta kaip kelių verčių logika, pavyzdžiui, produktai kartais turi būti padalinti į grupes pagal nustatytą kriterijų. Tačiau, nei dviejų verčių, nei keletu verčių logika yra tinkama susiduriant su netikslią arba neaiškia informacija ir jos apytiksliu samprotavimu, kaip neraiškios logikos teiginiai ir arba kiekybiniai arba funkciniai santykiai egzistuojantys daugelyje realaus pasaulio problemų.

Priešingai, neraiškios logikos rinkinių ir neraiškios logikos technikų taikymas leidžia koduoti sukonstruotą lingvistinę informaciją į skaitinę struktūrą. Taigi, netiksli arba neaiški informacija ir jos apytiksli samprotavimas gali būti apdorotas naudojantis matematiniais įrankiais. Neraiški logika gali būti apibūdinama kaip paprastos logikos pratesimas ar apibendrinimas. Iš tikrųjų, atvejo ar objekto „tiesa“, pavyzdžiui, lingvistinis teiginys gali būti aprašytas kaip atitinkama narių funkcija, esanti tėtinių verčių rinkinys,  $[0, 1]$ . Anksčiau paminėtame kokybinio valdymo pavyzdyje, jei mes sakome „produkcijos kokybė yra gera“, tai yra tipinis neraiškios logikos kalbos teiginys aprašantis kokybę. Norint išreikšti neraiškios logikos teiginį skaitine struktūra, neraiškios logikos narystė,  $m_A(x)$ , gali būti aprašyta taip, kad galėtų matuoti „tinkamumo laipsnį“ kiekvienam duotam produktui. Bendrąja prasme, narystės funkcija  $m_A(x)$  aprašo laipsnį kiek bet kuris  $x$  priklauso poaibiui  $A$ .

$$m_A(x) = \text{laipsnis}(x \in A) \quad (1.3)$$

Kitais žodžiais, narystės funkcijos reiškia sužymėjimą nuo  $x$  iki  $a$  vienetų intervalui,  $x \in [0, 1]$ , gali būti aprašytas kaip neraiškus rinkinys, kuomet sužymėjimas yra reglamentuotas kelių atvejo  $X$  netikslių savybių reikšminiu aprašymu. Tarkim

$$X = [x_1, x_2, \dots, x_n]^T \quad (1.4)$$

Neraiškus poaibis  $A$  gali būti žymimas nariais

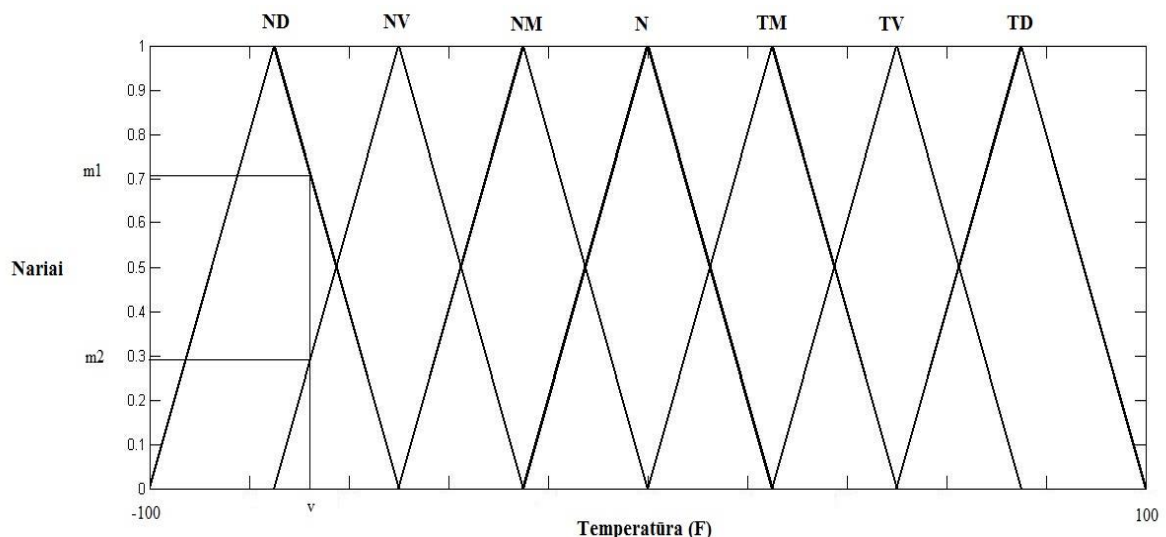
$$A = \sum_{i=1}^n m_A(x_i) x_i \quad (1.5)$$

$$m_A(x_i), i = 1, \dots, n$$

Jei  $X$  yra begalinis rinkinys,  $A$  gali būti aprašyta kaip

$$A = \sum_x m_A(x)x \quad (1.6)$$

Yra daugybė skirtingų formų, kurias pasitelkus aprašomos neraiškių kintamųjų rinkinių narių funkcijos. 4. Pav. parodo tipinę narių funkciją plačiai naudojamą praktikoje (ND – neigiamas didelis, NV – neigiamas vidutinis, NM – neigiamas mažas, N – nulis, TM – teigiamas mažas, TV – teigiamas vidutinis, TD – teigiamas didelis,  $m_1$  –  $v$  skaitinės reikšmės laipsnis, kuris priklauso neraiškiam rinkiniui NL,  $m_2$  –  $v$  skaitinės reikšmės laipsnis, kuris priklauso neraiškiam rinkiniui NM). Pirmas paveikslukas taip pat parodo, kad neraiškaus rinkinio elementai gali būti aprašyti atitinkamomis skaitinėmis reikšmėmis, kurių matematinis matavimas neraiškus kintamasis yra apribotas intervale  $[0, 1]$ . Taip pat, priešingai klasikiniams rinkiniams ir logikai, sutampančios sritys egzistuoja tarp apribojimų. Ši savybė padaro tikimybę  $P(A \subset A^C)$  nelygią nuliui, kur klasikiniame rinkinyje,  $P(A \subset A^C)$  yra visada lygi nuliui. Panašiai kaip tradiciniams rinkiniams, neraiškiems rinkiniams būdingos įvairios operacijos.



**1.3 pav.** Neraiškių rinkinių pavyzdinė narių funkcija.

Pagrindinės operacijos yra palyginimas, izoliavimas, sąjunga, susikirtimas, papildymas. Tarkim  $A$  ir  $B$  yra neraiškūs  $X$  poaibiai,  $X = \{x_i, i = 1, n\}$ , tai neraiškių rinkinių operacijos gali būti aprašomos:

1) Palyginimo:

$A$  ir  $B$  yra priskiriamos lygioms jei ir tik jei jų narių funkcijos yra vienodos,

$$m_A(x) = m_B(x), x \in X \quad (1.7)$$

2) Izoliavimo:

B izoliuoja A,

$$A \subset B, \text{ jei } m_A(x) \leq m_B(x), x \in X.$$

3) Sąjungos:

Sąjunga (atitinka „arba“) A ir B gali būti aprašyta

$$A \cup B, m_{A \cup B}(x) = \max(m_A(x), m_B(x)), x \in X \quad (1.8)$$

4) Sankirtos:

Sankirta (atitinka „ir“) A ir B gali būti aprašyta

$$A \cap B, m_{A \cap B}(x) = \min(m_A(x), m_B(x)), x \in X \quad (1.9)$$

5) Papildymo:

A papildymas, rašosi kaip  $A^c$  ir gali būti aprašytas

$$A^c = \{1 - m_A(x)\}, x \in X \quad (1.10)$$

Papildomai, neraiškaus rinkinio dydis gali būti išmatuotas sumuojant visas narių reikšmes, aprašytas kaip „sigma – count“:

$$\text{count}(A) = \sum_{x_i \in X} m_A(x_i), i = 1, \dots, n \quad (1.11)$$

Šis apibrėžimas susijęs su neraiškiu rinkiniu:

Išgautas neraiškus rinkinys:

Neraiškus rinkinys yra išgaubtas jei ir tik jei

$$m_A(px_1 + (1 - p)x_2) \geq \min(m_A(x_1), m_A(x_2)), x_1, x_2 \in X, p \in [0, 1] \quad (1.12)$$

6) Papildymo principas:

Tarkim  $f$  yra projektas iš  $\{x_i, i = 1, r\}$  į  $y$ ,

$$y = f(x_1, x_2, \dots, x_r)$$

tada

$$m_B(y) = \begin{cases} 0, & \text{jei } f'(y) = 0 \\ \sup_{x_1, \dots, x_r} \min(m_A(x_1), \dots, m_A(x_r)), & \text{visa kita} \end{cases} \quad (1.13)$$

7) Dekarto produktas:

$X_1, \dots, X_n$  Dekarto produktas gali būti išreikštas kaip

$$X = X_1 \times X_2 \times \dots \times X_n \quad (1.14)$$

Tarkim  $\{A_i \subset F(X_i), i = 1, n\}$ , tada Dekarto  $A_1, \dots, A_n$  produktas išreikštas

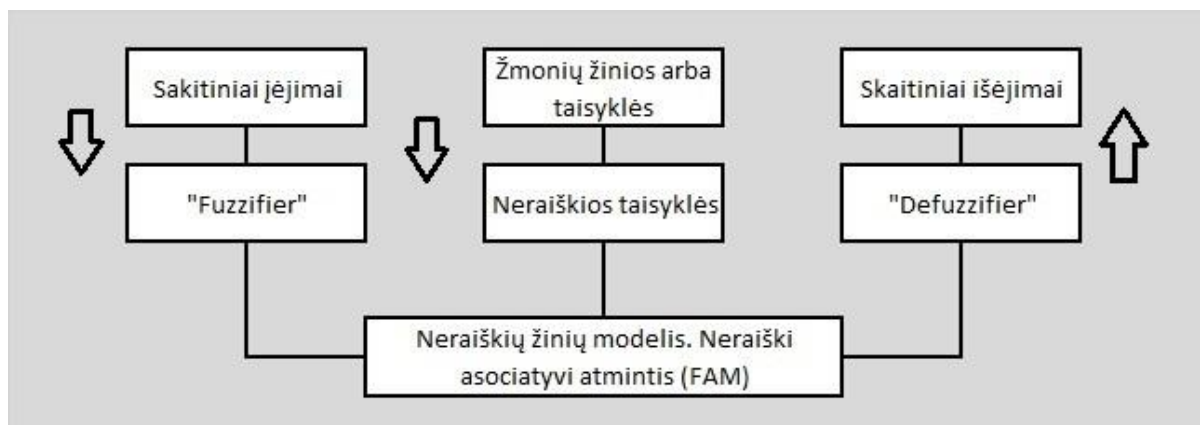
$$A = A_1 \times A_2 \times \dots \times A_n \quad (1.15)$$

Ir aprašomas

$$m_A(x_1, \dots, x_n) = \min(m_{A_1}(x_1), \dots, m_{A_n}(x_n)) \quad (1.16)$$

### 1.3 Neraiškių žinių modelis

Neraiškių žinių modelis gali būti naudojamas aprašant sistemos įėjimų/išėjimų elgesį arba atliekant sistemos valdymą ir sprendimų priėmimą. Neraiškių žinių modelio architektūra gali būti pavaizduojama 2.pav.



1.4 pav. Bendra neraiškių žinių modelio architektūra.

Neraiškiose sistemose, skaitinių įėjimų vertės pirma turi būti perskaičiuojamos į atitinkamas neraiškias vertes naudojant „Fuzzifiers“. Neraiškūs išėjimai tada paduodami iš neraiškaus modelio, kuris gali būti neraiškios logikos taisyklių rinkinys, neraiškūs sąryšiai arba net paprasta neraiški lentelė, su arba be gilaus neraiškaus apmąstymo. Galiausiai neraiškūs išėjimai gali būti perskaičiuojami atgal į jų tinkamas skaitines vertes (crisp) naudojant „Defuzzifiers“. Neraiškus modelis gali būti nustatytas remiantis sistemos neraiškiomis taisyklėmis ir / arba sistemos įėjimų/išėjimų duomenų neraiškiu identifikavimu. Neraiškios taisyklės dažniausiai surenkamos panaudojant žmogiškas žinias, arba jei būtina gali taip pat būti iširtos naudojantis neprižiūrimu mokymusi neuroniniuose tinkluose.

„Fuzzifier“ ir „Defuzzifier“ atlieka informacijos kodavimo ir šifravimo operacijas. Neraiškių modelių funkcijos yra gana panašios į ne – neraiškių sistemų matematiniu algoritmus. Kvantuojant bet kokią erdvę į septynias trapecijos formos sutampančias neraiškaus rinkinio vertes kaip parodyta 1. Pav. Tarkim, kad temperatūros intervalas T yra tarp [-100, 100], tada remiantis 1. Pav., „Fuzzifier“ operacijos yra gana paprastos, būtent, bet kuri skaitinė vertė yra paverčiama į jai atitinkantį neraiškų tašką nurodytame neraiškiam rinkinyje erdvėje. Tačiau, pasiekti „Defuzzifier“ operaciją yra labiau sudėtinga.

Paprasčiausias „Defuzzifier“ metodas yra parenkant elementą  $x_{max}$ , kuris turi minimalią narių funkcijos vertę išėjimų neraiškiam rinkinyje A:

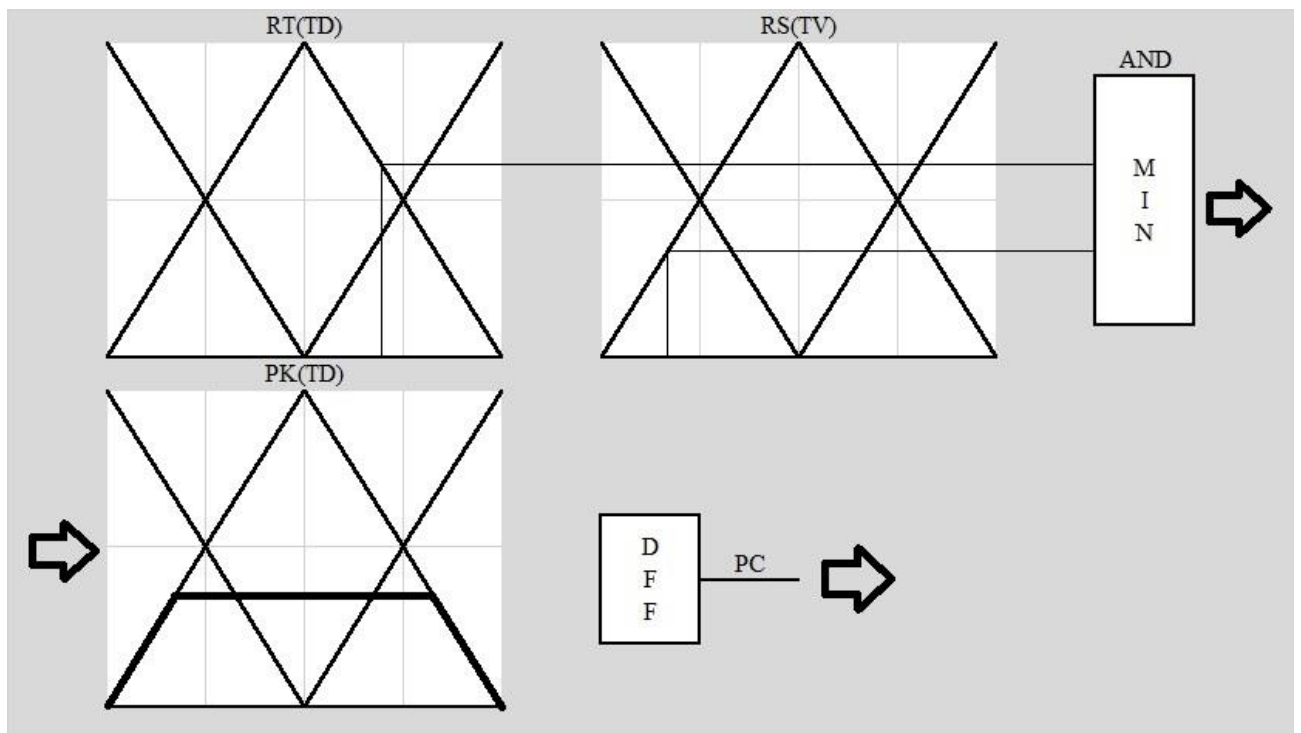
$$m_A(x_{max}) = \max\{m_A(x_i)\}, i = 1, n \quad (1.17)$$

Tačiau toks metodas turi du fundamentalius trūkumus. Pirma, A pasiskirstymo režimas nėra unikalus. Ši problema įtakoja koreliavimo minimumą koduojant labiau nei ji įtakoja koreliacijos produkto kodavimą. Antra, metodas ignoroja A informacijos bangų formą.

Alternatyva yra neraiškus centruotas „Defuzzifier“ metodas. Neraiškaus centroido išėjimo erdvės X atžvilgiu, ne – neraiški vertė, gali būti apskaičiuota:

$$\bar{A} = \frac{\sum_{i=1}^n x_i m_A(x_i)}{\sum_{i=1}^n m_A(x_i)} \quad (1.18)$$

Bendrai, neraiškus modelis gali būti aiškinamas kaip neraiškių įėjimų/išėjimų šablono kartografavimas arba neraiški asociatyvi atminti.



**1.5 pav.** Neraiškios taisyklės mechanizmo iliustracija.

(1.5 pav.) pavaizduoja, kaip gauti ne-neraiškias (crisp) PK vertes naudojant neraiškaus rinkinio kartografavimą (Jei neraiški taisyklė sujungta su „arba“, tada reikia „min“ pakeisti „max“). 3. Pav. atliekama taisyklė atrodo taip:

*Jei  $RT = TD$  IR  $RS = TV$ , tada  $PK = TD$*

Taigi, neraiški taisyklė aprašo neraiškų santykį tarp reakcijos temperatūros (RT), reakcijos slėgio (RS) ir produkto kompozicijos (PK).

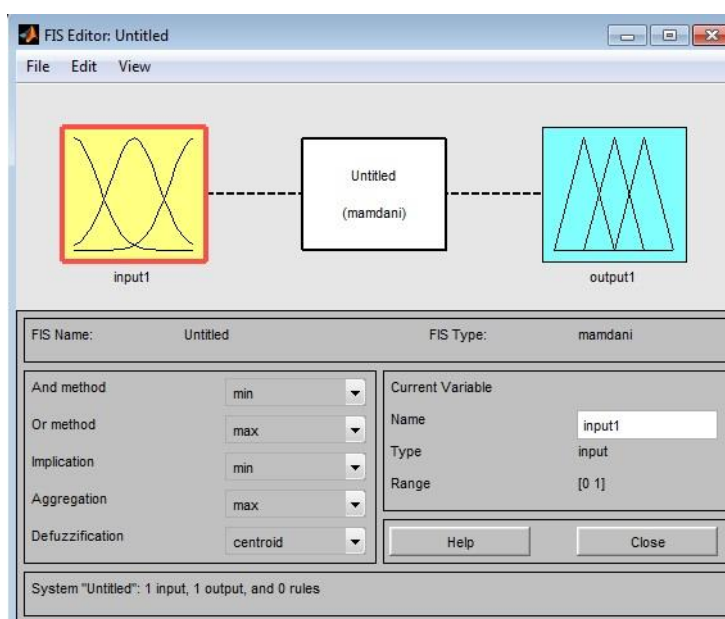


## 2. Programinės įrangos apžvalga

### 2.1 MATLAB Fuzzy Logic Toolbox

Fuzzy logic toolbox (FLT) grafinė sąsaja labai palengvina neraiškios logikos sistemų kūrimą ir manipuliavimą. FLT kaupia visą informaciją reikalingą neraiškių išvadų sistemai (Fuzzy inference system (fis)) vienoje matricoje, kuri vadinama „fis matrix“ ir, jei norima, gali būti išsaugota fis faile. Kurti naują neraiškių aibių sistemą arba keisti egzistuojančią sistemą galima labai paprastai, tiesiog keičiant fis matricą naudojant grafinę FLT sąsają.

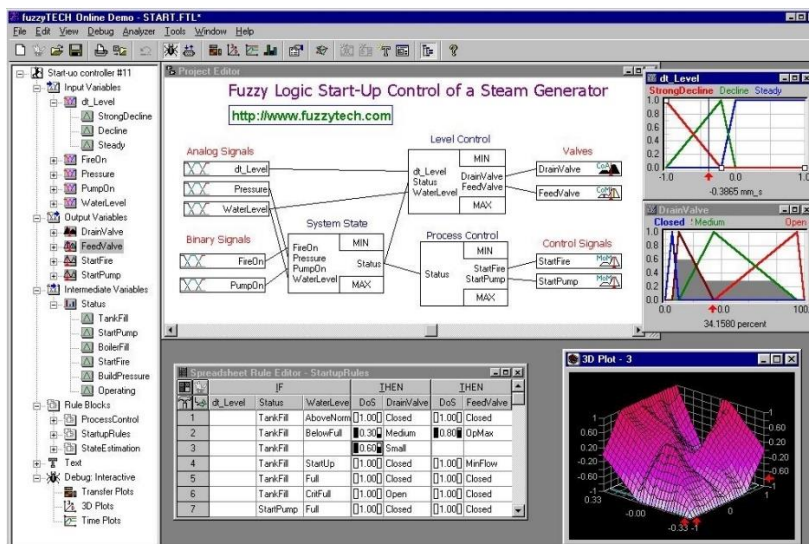
FLT gali būti panaudota ir iš komandinės eilutės, tačiau komandos labiau tinkamos sistemų kūrimui ir manipuliavimui neraiškių sistemų lygyje, nei neraiškių rinkinių lygyje. FLT sklandžiai veikia tiek su Mamdani, tiek su Takagi-Sugeno skaičiavimo tipais. Defuzikavimui naudoja apjungtų taisyklių svorio centro principą.



2.1 pav. MATLAB Fuzzy Logic Toolbox

### 2.2 FuzzyTECH

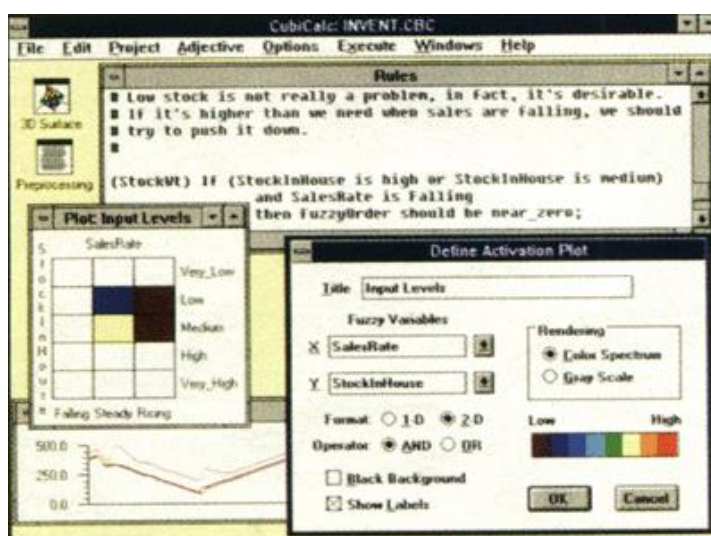
fuzzyTECH yra neraiškių aibių logikos sistemų kūrimo įrankis. Jis naudoja panašų stilių ir valdymą kaip Microsoft Windows ir Microsoft Office programiniai produktai. Pagrindinis langas įgalina struktūrinį priėjimą prie visų neraiškių aibių logikos sistemos komponentų projekte tokiu pačiu būdu kaip ir Windows Explorer leidžia vartotojui naršyti struktūrą vartotojo kompiuteryje. Redaktoriaus ir analizatoriaus langai leidžia projektuoti kiekvieną neraiškių aibių logikos sistemos komponentą grafiškai. Defuzikavimui naudoja apjungtų taisyklių svorio centro principą.



2.2 pav. FuzzyTECH

### 2.3 CubiCalc

CubiCalc paketas apima interaktyvią aplinką skirtą ekspertinių sistemų ir neraiškių aibių logikos sistemų kūrimui, taip pat tuo pat metu vartotojui leidžia registruotis prie sistemos kaip atskiros sistemos. Iš kitų programinių paketų CubiCalc galima išskirti dėl gerai veikiančio plėtinio „Rule Maker“, kuris leidžia išspręsti vieną iš didžiausių problemų dirbant su neraiškių aibių logika – automatinis neraiškių aibių taisyklių kūrimas. Šią funkciją taip pat turi ir MATLAB fuzzy logic toolbox. Defuzikavimui naudoja apjungtų taisyklių svorio centro principą. Defuzikavimui naudoja apjungtų taisyklių svorio centro principą.



2.3 pav. CubiCalc

## 2.4 FuzzyCLIPS

FuzzyCLIPS yra neraiškių aibių logikos CLIPS(C Language Integrated Production System) ekspertinės sistemos paketo iš NASA plėtinys. Jis buvo sukurtas Kanados Nacionalinės Tyrėjų Tarybos (National Research Council of Canada) Informacinių Technologijų Instituto (Institute for Information Technology) Integruoto Samprotavimo Grupės (Integrated Reasoning Group) ir buvo plačiai platinamas daugelį metų. Šis plėtinys pagerina CLIPS suteikdamas neraiškaus samprotavimo galimybę, kuris yra pilnai integruotas su CLIPS faktais ir išvadų mechanizmu, leidžianti aprašyti ir manipuluoti neraiškius faktus ir taisykles. FuzzyCLIPS gali susidoroti su tiksliais, neraiškiais arba apjungtu samprotavimu, taip leidžiantis neraiškioms ir normalioms sąlygoms būti laisvai maišomoms ekspertinės sistemos taisyklėse ir faktuose. Sistema naudoja du pagrindinius netikslumo idėjas, neaiškumas ir neapibrėžtumas. Jis turi naudingą aplinką skirtą neraiškių aibių logikos aplinkų kūrimui, tačiau reikalauja labai daug pastangų atnaujinti ir palaikyti kai naujos CLIPS versijos yra išleidžiamos. Defuzikavimui naudoja apjungtų taisyklių svorio centro principą. Taip pat neturi grafinės sąsajos.

### 3. Sukurtas programinis paketas FuzzyThis (FT)

Sukurtas programinis paketas leidžia racionaliau įvertinti neraiškių aibių sistemos išėjimą, kadangi neraiškumo eliminavimas atliekamas ne pagal svorio centro metodą, bet įvertinant išėjimo kintamojo neapibrėžtumą. Taip pat leidžia realizuoti hierarchines neraiškių aibių sistemų struktūras. Paketas yra autonomiškas ir gali būti įtrauktas į kitas skaičiavimo procedūras ir procesų valdymo sistemas.

#### 3.1 FT neraiškių aibių sistemos įėjimų /išėjimų funkcinės priklausomybės

- Taisyklių sistema: *JEI  $x_l$  yra  $F_n^l$  TAI  $y$  yra  $G^l$*

$x_l$  – pasirinktas įėjimas.

$F_n^l$  – pasirinkto įėjimo pasirinkta narystės funkcija.

$y$  – išėjimas.

$G^l$  – pasirinkta išėjimo narystės funkcija.

- Narystės funkcijos tenkinimas:

$$\mu_{F_i^l}(x_i) = \exp\left(\frac{-(x_i - r_i^l)^2}{2 * ((\sigma_i^l)^2 + \varepsilon)}\right), \mu_{G_i^l}(y) = \exp\left(\frac{-(x_i - s^l)^2}{2 * ((\delta^l)^2 + \varepsilon)}\right) \quad (3.1)$$

$x_i$  - įėjimo duomenys

$r_i^l$  – įėjimo narystės funkcijos vidurkis

$\sigma_i^l$  – įėjimo narystės funkcijos neraiškumas

$y$  – išėjimas

$s^l$  – išėjimo narystės funkcijos vidurkis

$\delta^l$  – išėjimo narystės funkcijos neraiškumas

- Galutinis išėjimas:

$$y = \frac{\sum_{l=1}^M s^l \left[ \prod_{i=1}^n \mu_{F_i^l}(x_i) \right] / [(\delta^l)^2 + \varepsilon]}{\sum_{l=1}^M \left[ \prod_{i=1}^n \mu_{F_i^l}(x_i) \right] / [(\delta^l)^2 + \varepsilon]} \quad (3.2)$$

$n$  – taisyklės įėjimų skaičius

$M$  – taisyklių skaičius

$\varepsilon$  – mažas skaičius dėl skaičiavimo procedūros stabilumo (0,001)

### 3.2 FT programavimo terpė ir kalba.

Programuoti paketa pasirinkau „Microsoft Visual Studio 2015“ programavimo terpėje. Visual Studio yra pilnas programavimo įrankių rinkinys skirtas kurti ASP.NET tinklalapius, XML tinklalapių paslaugas, stalinių kompiuterių programas, mobilias programas. Visual Basic, Visual C#, ir Visual C++ visos šios kalbos naudoja tą pačią integruotą programavimo aplinką (IDE), kuri leidžia įrankių dalinimąsi ir palengvina apjungtų kalbų programavimo problemų sprendimą. Papildomai, šios kalbos naudojami .NET struktūra, kuri teikia prieigą prie pagrindinių technologijų taip supaprastindama ASP Web, XML Web programų kūrimą.

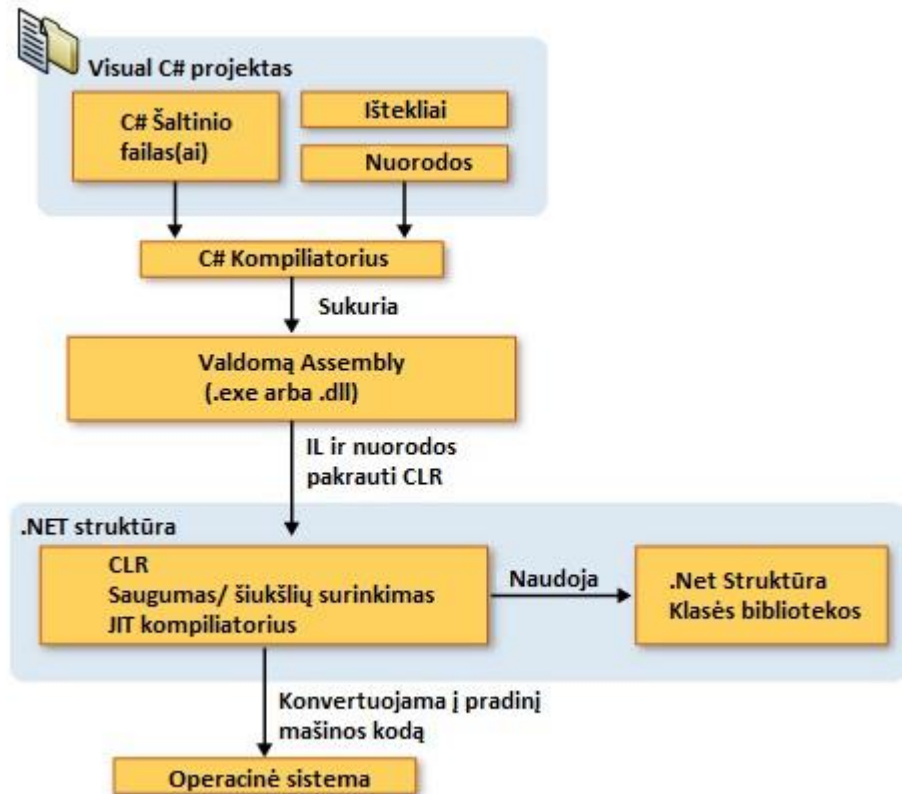
Programavimo kalbą pasirinkau Visual C#, kadangi esu susipažinęs su C# kalbos sintakse ir struktūra. C# sintaksė yra labai išraiškinga, tačiau tuo pačiu paprasta ir lengvai išmokstama. Ji supaprastina daugybę C++ sunkumų ir teikia gerus sprendimus kaip nulinės vertės, rodikliai, delegatai ir tiesioginis atminties pasiekimas, kurių nerasite Java kalboje. C# palaiko bendrus metodus ir tipus, kurie teikia padidintą įvedimo saugumą ir našumą, ir iteratorius, kurie įgalina rinkinio klasių įgyvendinimą norint aprašyti netradicinį iteracinį veiksmą, kuris gali būti paprastai panaudotas kliento kode.

Kaip objektinė kalba C# palaiko „encapsulation“, paveldėjimo ir polimorfizmo sąvokas. Visi kintamieji ir metodai, įskaitant ir Main metodą, programos įėjimą, yra išsaugomi klasės aprašymuose. Klasė gali paveldėti tiesiogiai iš pirminės klasės, tačiau taip ji gali aprašyti bet kokią sąsajos skaičių. Metodai, kurie nepaiso virtualių metodų pagrindinėje klasėje reikalauja „override“ raktažodžio kaip būdo išvengti atsitiktinio esamo metodo apibrėžimo.

C# programa paleidžiama ant .NET struktūros, kaip vientisas Windows komponentas, kuris apima ir virtualią paleidimo sistemą vadinamą „common language runtime“ (CLR) ir yra vieningas klasių bibliotekos rinkinys. CLR yra komercinis „common language infrastructure“ (CLI) Microsoft sprendimas, internacionalinis standartas, kuris yra pagrindas kuriant paleidimo ir programavimo aplinkas, kuriose kalbos ir bibliotekos veikia sklandžiai kartu.

Pirminis kodas parašytas C# kalba kompiliuojamas į tarpinę kalbą (IL - intermediate language), kuri atitinka CLI specifikaciją. IL kodas ir ištekliai, kaip bitmaps ir strings, yra išsaugomi faile vadinamame „assembly“ ir dažniausiai su pratesimu .exe arba .dll. Assembly savyje turi sąrašą, kuris teikia informaciją apie jo tipus, versiją, kultūrą, saugumo reikalavimus.

Kuomet c# programa yra paleidžiama, assembly yra pakraunamas į CLR, kuris gali imtis įvairių veiksmų remiantis informacija sąrašė. Tada, jei saugumo reikalavimai yra tenkinami, CLR atlieka JIT(just in time) kompiliaciją norint konvertuoti IL kodą į paprastas mašinos instrukcijas. CLR taip pat teikia kitas paslaugas susietas su šiukšlių surinkimu, išimčių tvarkymu, ir resursų valdymu.

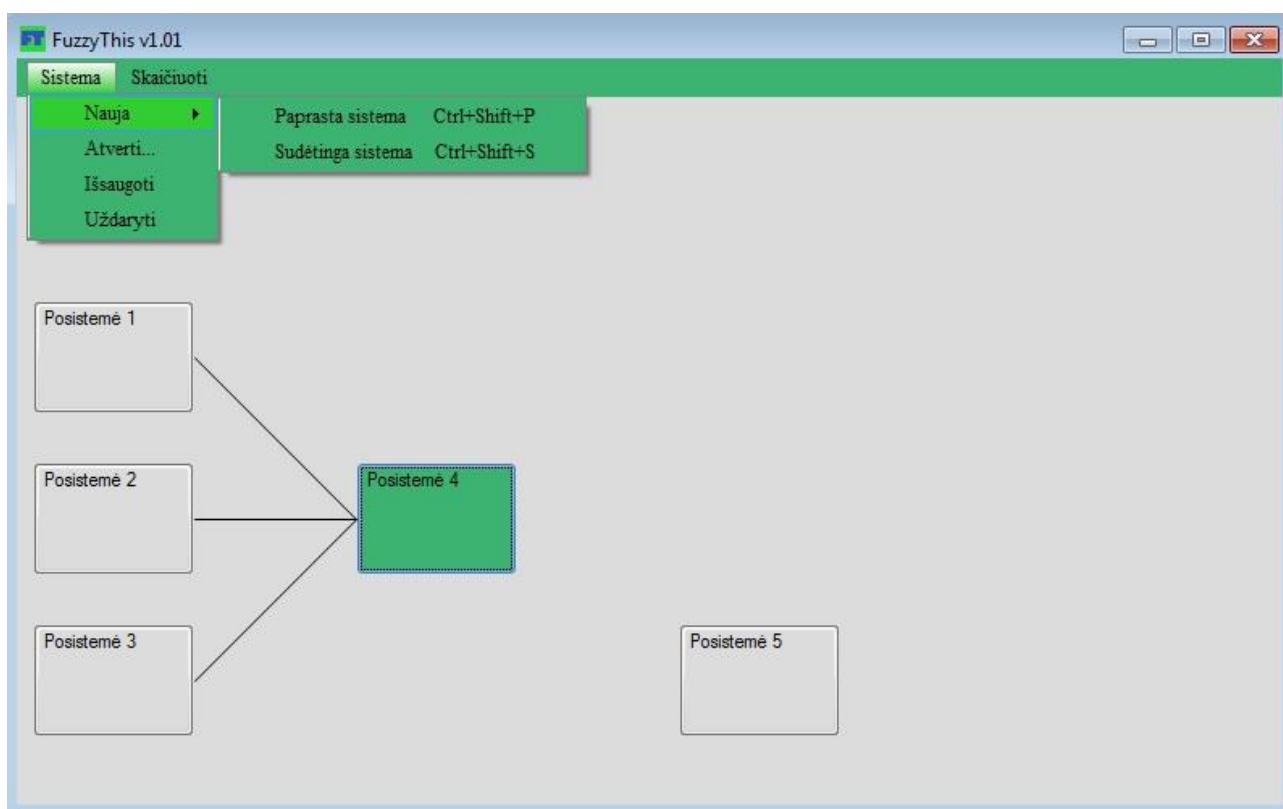


**3.1 pav.** .NET struktūros platformos architektūra.

Programavimo kalbos sąveika yra pagrindinė .Net struktūros ypatybė. Kadangi IL kodas pagamintas C# kompiliatoriaus atitinka „Common Type Specification“ (CTS), IL kodas generuotas iš C# gali sąveikauti su kodu, kuris buvo generuotas iš Visual C#, Visual C++, Visual Basic arba bet kurios iš daugiau nei dvidešimties kitų CTS-kompilijuojamų kalbų .NET versijos. Viena assembly gali būti sudaryta iš daugelio modulių parašytų skirtingomis .NET kalbomis, ir tipais gali nurodyti vienas kitą lyg visi būtų parašyti tą pačia kalba.

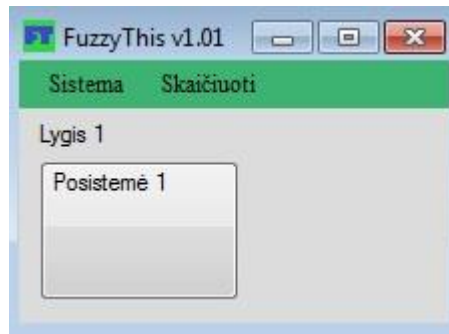
Papildomai paleidimo paslaugoms, .NET struktūra tai pat turi vis pasipildančią biblioteką, kurios dydis siekia virš keturių tūkstančių klasių organizuotų pagal namespace, kad teiktų didelę funkcionalumo įvairovę viskam nuo failų įvedimo ir išvedimo iki tekstinio kintamojo manipuliavimo bei Windows Forms valdymo.

### 3.3 FT vartotojo sąsaja

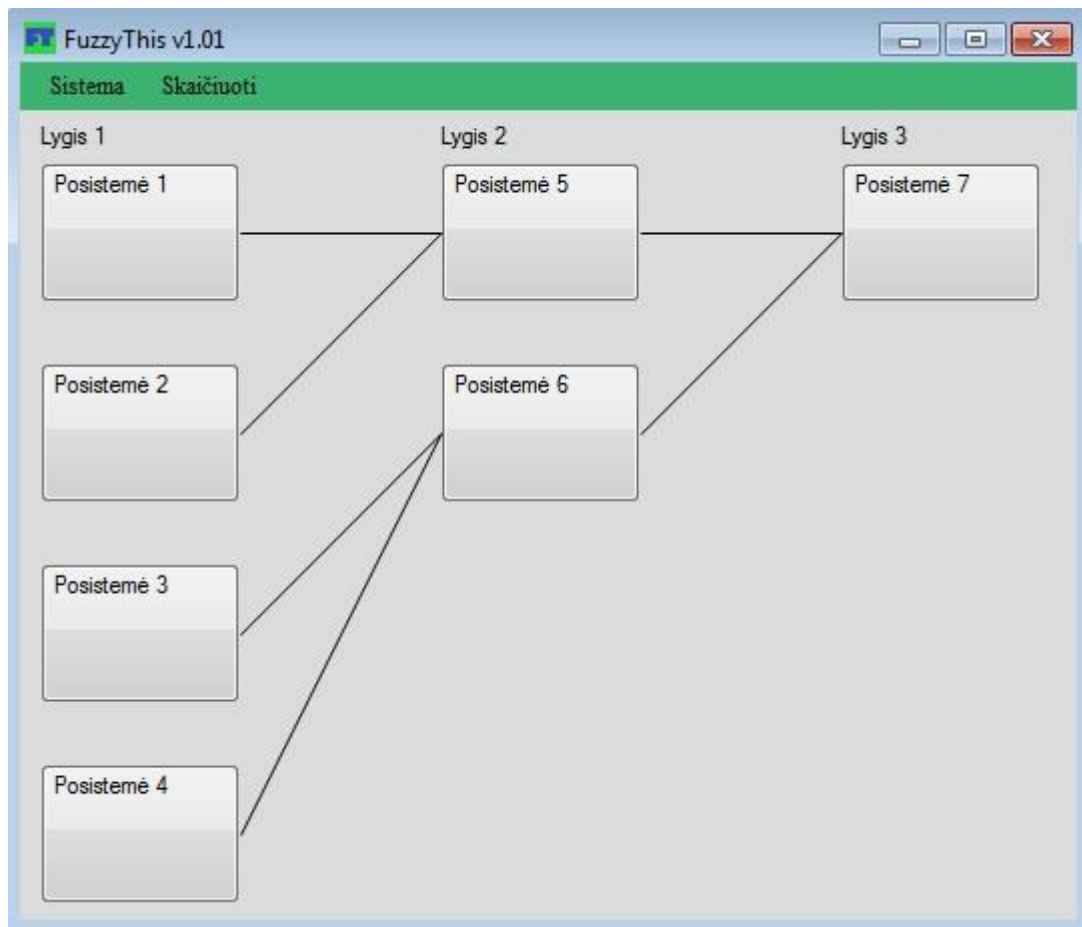


3.2 pav. FT vartotojo sąsaja

Programos langas, pavaizduotas 3.2 paveikslėlyje atsidaro, kuomet paleidžiama programa, be jokių papildomų dialogų. Pagrindinio lango meniu juostoje yra du pasirinkimai „Sistema“ ir „Skaiciuoti“. Paspaudus pasirinkimą „Sistema“ atsiveria šio pasirinkimo konteksto meniu, kurį sudaro pasirinkimai: „Nauja“, „Atverti...“, „Išsaugoti“, „Uždaryti“. Pelės mygtuku paspaudus pasirinkimą „Nauja“ atsiveria dar du šio meniu pasirinkimai: „Paprasta sistema“ ir „Sudėtinga sistema“. Šie pasirinkimai taip pat gali būti aktyvuoti ir specialiomis mygtukų kombinacijomis (Ctrl+Shift+P – „Paprasta sistema“ ir Ctrl+Shift+S – „Sudėtinga sistema“). Būtina paminėti, kad mygtukų kombinacijos veikia tik pažymėjus elementą arba kuomet langas tuščias, pavyzdžiui, norint pridėti posistemę prie jau esamų posistemų, pasinaudojus mygtuko kombinacija, reikia pirma pažymėti jau esamą mygtuką, priešingai, jei langas tuščias - pažymėjimas nebūtinai. Šios instrukcijos reikalingos tik mygtuko kombinacijoms. Paspaudus pasirinkimą „Paprasta sistema“ meniu lange atsiranda viena posistemė su hierarchinio lygio pavadinimu, pavaizduota paveikslėlyje 3.3 pav. Paspaudus pasirinkimą „Sudėtinga sistema“ meniu lange atsiranda septynios posistemės, sujungtos į bendrą



**3.3 pav.** „Paprasta sistema“

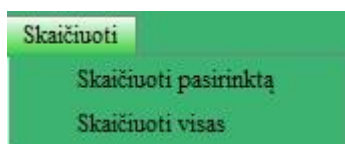


**3.4 pav.** „Sudėtinga sistema“ FT aplinkoje

hierarchinę struktūrą, su trimis hierarchinių lygių pavadinimais, pavaizduota paveikslėlyje 3.4. Pasirinkimai „Paprasta sistema“ ir „Sudėtinga sistema“ Gali būti įvykdyti tik kuomet pagrindinis langas tuščias. „Atverti...“ pasirinkimas atidaro Windows Explorer dialogą, kuriame gali naršyti kompiuterio failus ir rasti atitinkamą išsaugotą sistemą. Dialogas atpažins netinkamą failą, tačiau jei failas yra tinkamas ir jo formatavimas yra blogas – programa nustos funkcionavus. Tokiu atveju reikia įsitikinti, ar tinkamas failas buvo pažymėtas ir ar pažymėtas failas gerai suformatuotas. „Išsaugoti“ pasirinkimas įrašo visus reikalingus sistemos duomenis iš duomenų masyvų į vieną .txt failą, kurį galima atkurti kitą



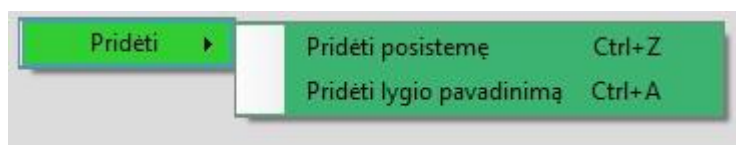
kartą paleidus programą. „Uždaryti“ uždaro programą ir ištrina visus neišsaugotus duomenis. Meniu juostos pasirinkimas „Skaičiuoti“ atsiveria dar du pasirinkimai: „Skaičiuoti pasirinktą“, „Skaičiuoti visas“.



**3.5 pav.** „Skaičiuoti“ pasirinkimas FT

„Skaičiuoti pasirinktą“ leidžia atlikti skaičiavimo veiksmą ant paspaustos posistemės (Posistemė paspaudžiama po šio pasirinkimo paspaudimo). Šis veiksmas praktiškesnis, nei „Skaičiuoti visas“, kadangi galima skaičiuoti norimas posistemės dar nebaigus visos sistemos ir taip išvengti klaidų ar eiliškumo problemų. Programa turi apsaugas nuo šių veiksmų netinkamo pasirinkimo, tačiau nėra testuota vartotojų. Atliekant šį veiksmą hierarchinėms struktūroms būtina skaičiuoti iš žemesnio lygio į aukštesnį, kadangi tokiu atveju aukštesnio hierarchinio lygio sistemos neturės duomenų ir programa gali sustoti veikti arba skaičiavimai bus nevykdomi. „Skaičiuoti visas“ skaičiuoja visas posistemės iš viršaus į apačią ir iš kairės į dešinę.

Pagrindiniame lange nebūtina naudotis „Nauja“ pasirinkimu norint pridėti posistemės, šis pasirinkimas skirtas kaip greitas sistemos sukūrimas. Paspaudus kairį pelės klavišą bet kurioje (tuščioje vietoje) programos lange atsidaro meniu, pavaizduotas 3.6 paveikslėlyje.



**3.6 pav.** „Pridėti“ pasirinkimas FT aplinkoje

Užvedus pelės rodyklę ant „Pridėti“ meniu pasirinkimo atsiranda dar du pasirinkimai: „Pridėti posistemę“, „Pridėti lygio pavadinimą“. Abu pasirinkimai turi mygtuko kombinacijas (Ctrl+Z – prideda posistemę, Ctrl+A – prideda hierarchinio lygio pavadinimą).

Paspaudus „Pridėti lygio pavadinimą“ veiksmą truputį žemiau pagrindinio programos lango meniu juostos, kur atliktas pelės paspaudimo įvykis atsiranda lygio pavadinimas. Visi elementai pagrindiniame programos lange turi lokalias koordinatas. Kiekvienas elementas gali būti kilnojamas, tačiau padėjus elementą jam apskaičiuojamos koordinatės, kad būtų išlaikomas atstumas tarp elementų. Hierarchinio lygio pavadinimas čia turi išimtį, kadangi jis gali judėti tik horizontaliai. Pridėjus lygio pavadinimą toje pačioje vietoje jis keliauja per atitinkamą atstumą į dešinę. Kilnojant lygio pavadinimą

ir jį padedant ant kito lygio pavadinimo – jie susikeičia vietomis. Du kartus spragtelėjus ant lygio pavadinimo atsidaro pervadinimo dialogas, kuriame galima pakeisti lygio pavadinimą, tačiau įvedamų simbolių skaičius yra ribojamas. Pervadinimas pavaizduotas paveikslėlyje 3.7 pav.

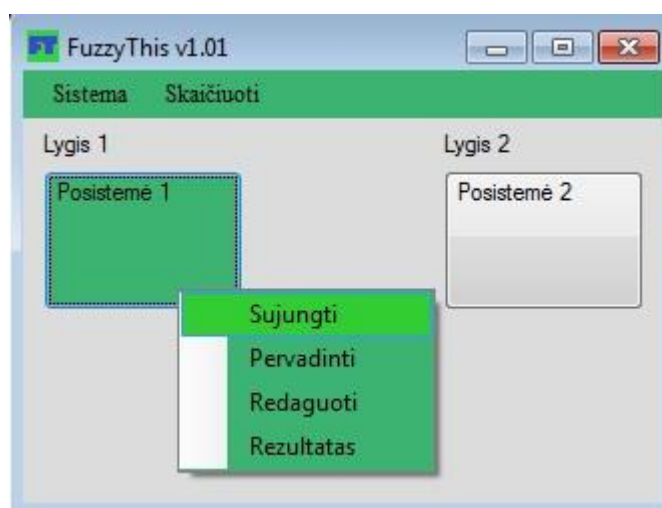


**3.7 pav.** „Pervadinti“ pasirinkimas FT aplinkoje

Dešiniu pelės klavišu paspaudus ant lygio ir tada paspaudus „Delete“ mygtuką klaviatūroje – lygis ištrinamas.

Paspaudus „Pridėti posistemę“ pelės paspaudimo vietoje atsiranda kvadratinis elementas kuris atlieka vizualią posistemės/sistemos funkciją. Posistemė perstumama į viršų iki pat viršaus arba iki kitos posistemės, pelė nustato tik horizontalią vietą pagrindiniame programos lange. Posistemę galima kilnoti ne tik horizontaliai bet ir vertikalčiai. Padėjus ant kitos posistemės – jos susikeičia vietomis. Pažymėjus posistemę dešiniu pelės klavišu ir paspaudus „Delete“ mygtuką klaviatūroje – posistemė ištrinama.

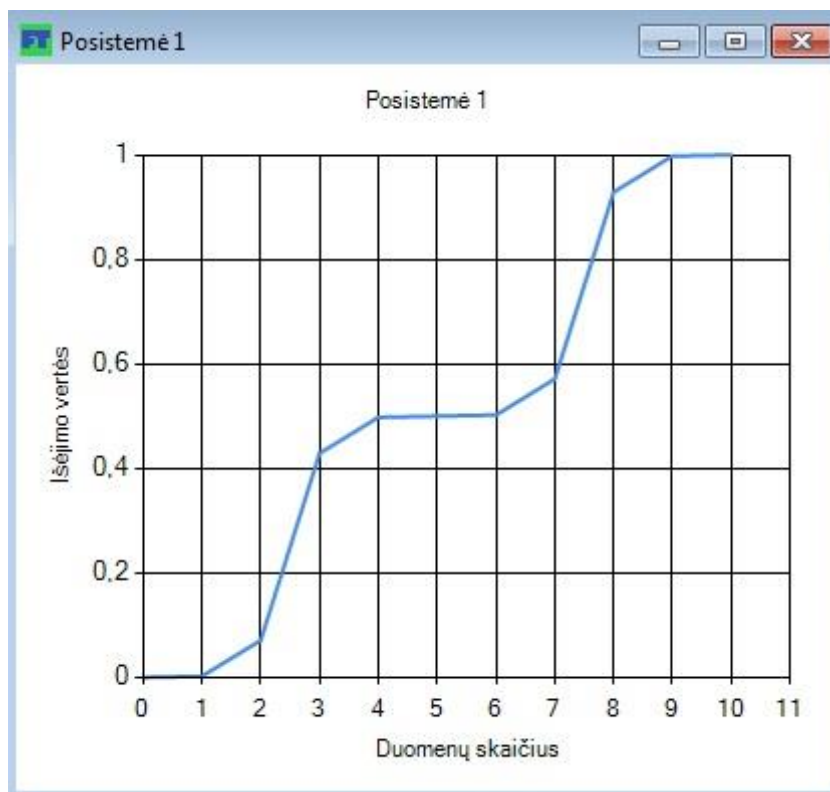
Posistemės elementas turi savo nuosavą meniu, kuris iškviečiamas ant jo paspaudus kairį pelės klavišą. Meniu pavaizduotas paveikslėlyje 3.8



**3.8 pav.** Posistemės meniu FT aplinkoje

Šis meniu turi keturis veiksmus: „Sujungti“, „Pervadinti“, „Redaguoti“, „Rezultatas“. Visi šie veiksmai leidžia atlikti atitinkamą veiksmą pasirenkamai posistemei (pirma pasirenkamas veiksmas tada pasirenkama posistemė). „Sujungti“ veiksmas leidžia sujungti posistemės tiesia, veiksmas atliekamas

paspaudžiant šį veiksma meniu ir tada paspaudžiant pirmą posistemę ir po to antrą posistemę tarp kurių norite nubrėžti tiesę ir jas sujungti į hierarchinę sistemą. Sujungimas turi tam tikras taisykles, tiesė brėžiama iš žemesnio į aukštesnį lygį. Kitais atvejais programa nepieš tiesės ir įspės vartotoją apie neteisingą veiksma. Posistemės sujungtos tiesėmis jau nebegali būti perkeltamos, nebent sujungimas būtų panaikintas. Ištrinti tiesę galima laikant paspaudus kairį „Ctrl“ klavišą ir spragtelėjus kairiuoju perės klavišu ant tiesės, kurią norima ištrinti. „Pervadinti“ veiksmas atliekamas tokiu pat principu kaip ir lygio pavadinimui. „Rezultatas“ veiksmas leidžia atidaryti suskaičiuotų duomenų grafinę sąsają, kur galima įvertinti atliktus skaičiavimus. Rezultatas pavaizduotas paveikslėlyje 3.9.



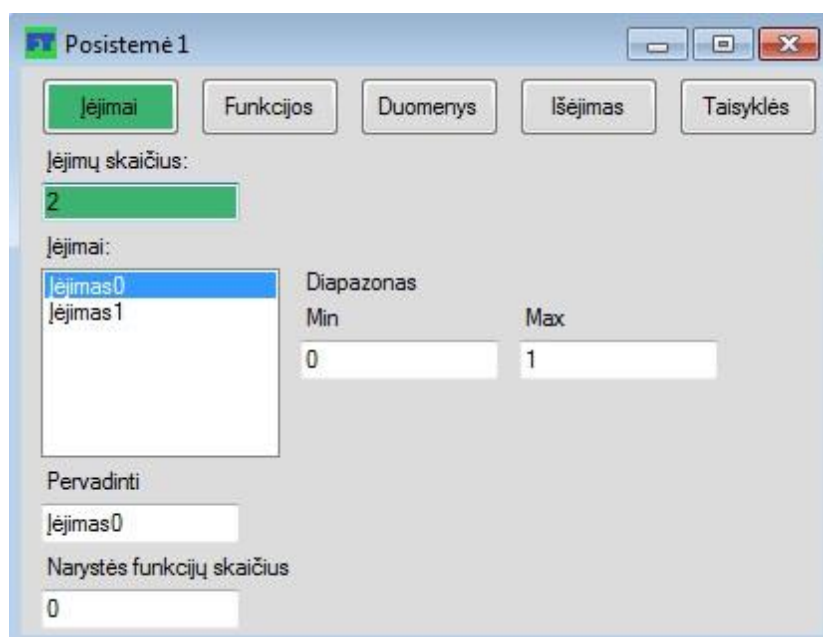
**3.9 pav.** „Rezultatas“ pasirinkimas FT aplinkoje

„Redaguoti“ veiksmas atidaro posistemės dialogą, kuriame galima aprašyti ir redaguoti jau aprašytą posistemę.



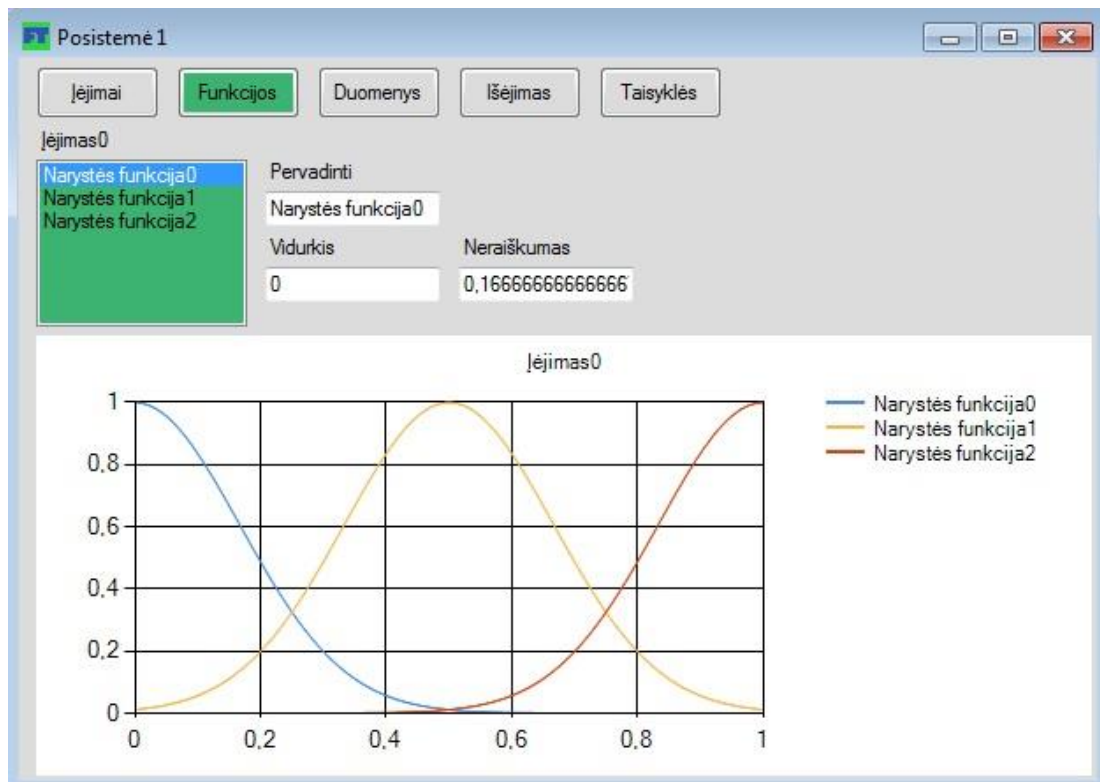
**3.10 pav.** Posistemės dialogas FT aplinkoje

Posistemės dialogas susideda iš penkių skirtingų sričių: „Įėjimai“, „Funkcijos“, „Duomenys“, „Išėjimas“, „Taisyklės“. Kiekvienoje srityje aprašomas tam tikra posistemės duomenų ar parametrų dalis. Sritčių mygtukus galima spausti laisvai, tačiau yra tam tikra eiliškumo tvarka, pavyzdžiui, taisyklių sritis neatsidarys, jei įėjimų skaičius mažesnis nei vienetas, ši taisyklė taip pat galioja ir „Funkcijos“ sritčiai. Duomenų sritį galima atsidaryti bet kada, tačiau patarčiau duomenis įkelti jau aprašytai sistemai, kadangi taip išvengiama nemažai klaidų keičiant įėjimų skaičių ar hierarchinės sistemos struktūrą. Taigi, praktiškiausias įvedimo eiliškumas būtų: aprašyti „Įėjimai“ sritį, tada kiekvienam įėjimui aprašyti „Funkcijos“ sritį, tada „Išėjimas“ ir galiausiai „Taisyklės“ sritį. Tada, jei sistema nėra hierarchinė – įkelti įėjimų duomenis, kitu atveju aprašyti ir kitas posistemas ir tik tada įkelti reikalingus duomenis, ir galiausiai skaičiuoti.



**3.11 pav.** „Įėjimai“ pasirinkimas FT aplinkos posistemės

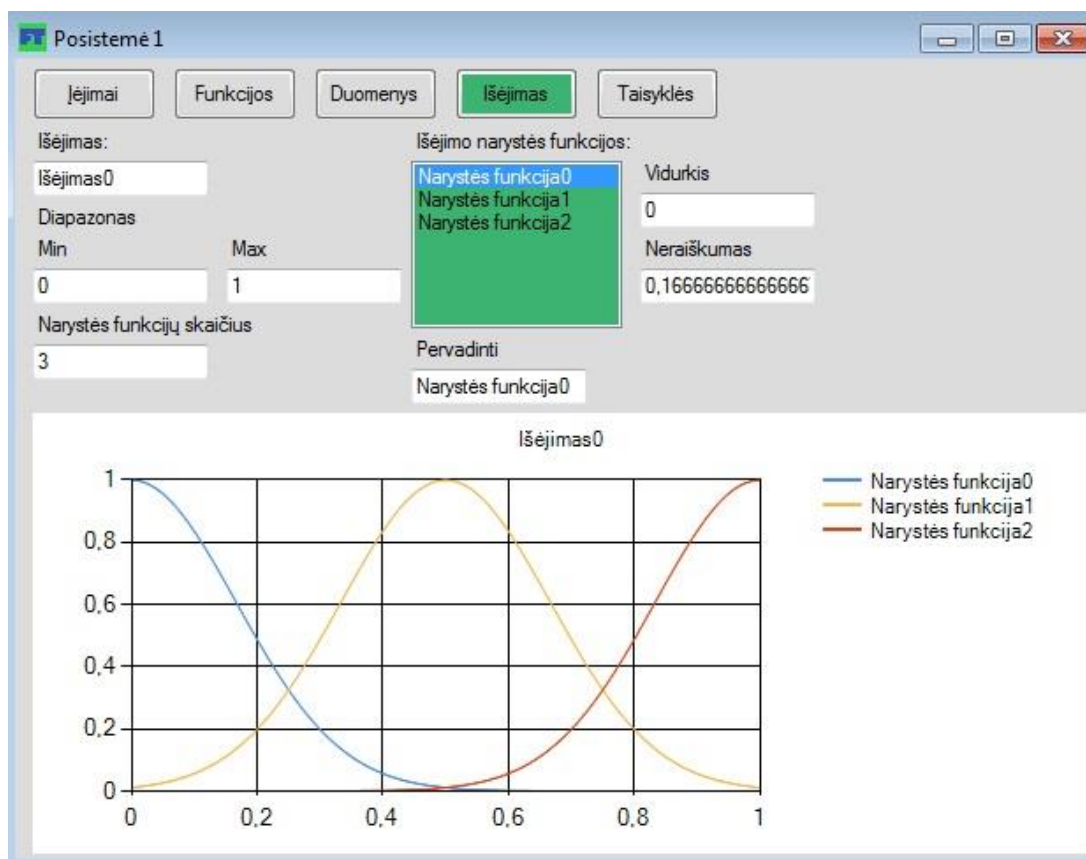
Įėjimo srityje aprašomi įėjimų skaičius, jei neįrašytas lygus nuliui, kiekvieno įėjimo diapazonas, kuris įvedant įėjimų skaičių automatiškai nustatomas nuo nulio iki vieneto, pavadinimas, kuris taip pat automatiškai nustatomas kaip įėjimas(įėjimo numeris) ir narystės funkcijų skaičius, kuris automatiškai nustatomas 0. Šiuos visus parametrus galima keisti ir aprašyti norimus posistemės įėjimus. Pasirinkus įėjimą iš „Įėjimai:“ pasirinkimų lentelės galima spausti mygtuką „Funkcijos“. Šį veiksmą pakartoti kiekvienam įėjimui, kurį norima keisti, tačiau tai nėra būtina, nes programa automatiškai apskaičiuoja narystės funkcijos parametrus remiantis įėjimo diapazonu ir narystės funkcijų skaičiumi.



3.12 pav. „Funkcijos“ pasirinkimas FT aplinkos posistemės dialoge

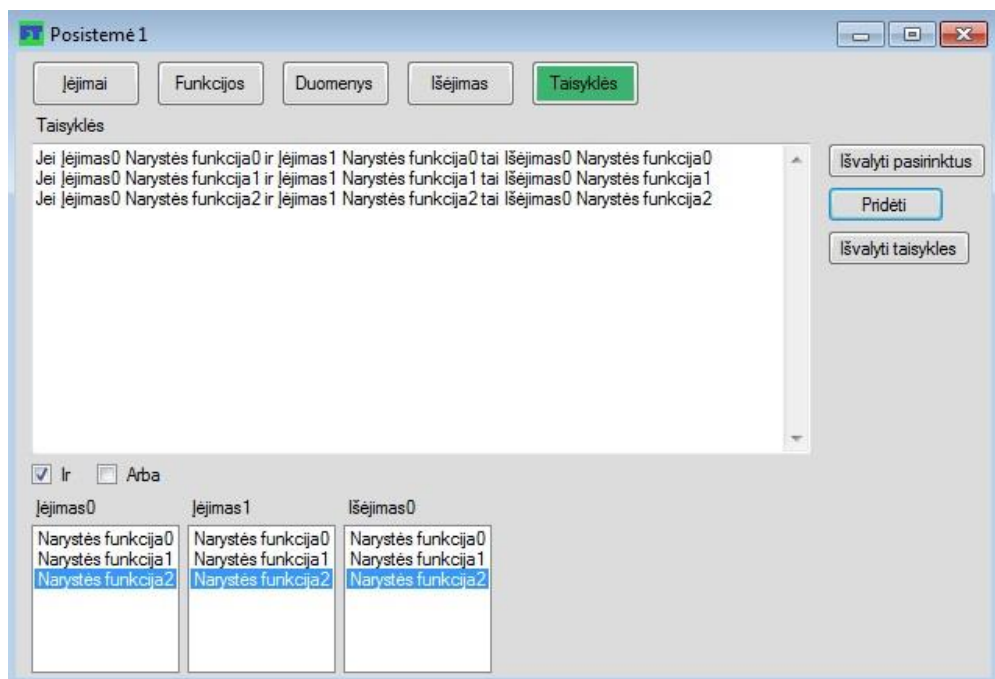
„Funkcijos“ srities lange atvaizduojamos automatiškai sugeneruotos pasirinkto įėjimo narystės funkcijos. Šios srities lange galima tiek grafiškai keisti įėjimo narystės funkcijų parametrus, tiek juos įvesti į aukščiau grafiko esančius teksto langelius. Galima pervadinti kiekvieną įėjimo narystės funkciją, aprašyti jos vidurkį ir neraiškumą. Yra tam tikros taisyklė ribojančios vertės, kad įvestos vertės neviršytų ribų, todėl yra praktiškiau naudoti grafinį funkcijų manipuliavimą, nebent norima pasiekti labai didelį narystės funkcijos tikslumą. Grafiškai funkcijos manipuluojamos pažymint funkciją grafike ir laikant paspaudus kairį pelės klavišą velkant pelę vertikaliai aukštyn arba žemyn, norint keisti neraiškumą (plotį) arba velkant pelę horizontaliai į kairę arba dešinę, norint keisti narystės funkcijos vidurkį (centro padėtį).

„Išėjimas“ sritis aprašoma tokiu pat būdu kaip ir įėjimai, tik kadangi posistemėje yra vienas išėjimas, funkcijos sritis nebūtinai turi būti atskira. Išėjimo aprašymas pavaizduotas paveikslėlyje 3.13.



**3.13 pav.** „Išėjimas“ pasirinkimas FT aplinkos posistemės

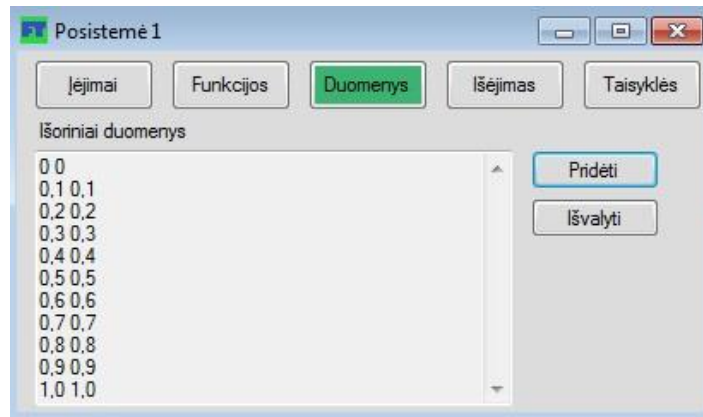
„Taisyklės“ sritį galima redaguoti kuomet įėjimai ir išėjimas yra aprašyti. Kai taisyklių redaktoriuje yra bent viena taisyklė – įėjimai ir išėjimas nebus leidžiami koreguoti. Tada vienintelis aktyvus mygtukas bus „Duomenų“. Aprašytos taisyklės pavaizduotos paveikslėlyje 3.14.



**3.14 pav.** „Įėjimai“ pasirinkimas FT aplinkos posistemės

Taisyklių srities lange yra įėjimų narystės funkcijų ir išėjimo narystės funkcijų pasirinkimo lentelės, „Ir“ bei „Arba“ pasirinkimo laukeliai, mygtukai: „Pridėti“, „Išvalyti taisykles“ ir „Išvalyti pasirinktus“. Iš eilės mygtukai atlieka funkcijas: prideda iš pasirinkimų sudarytą taisyklę, išvalo taisykles, išvalo pasirinktus elementus iš pasirinkimo lentelių.

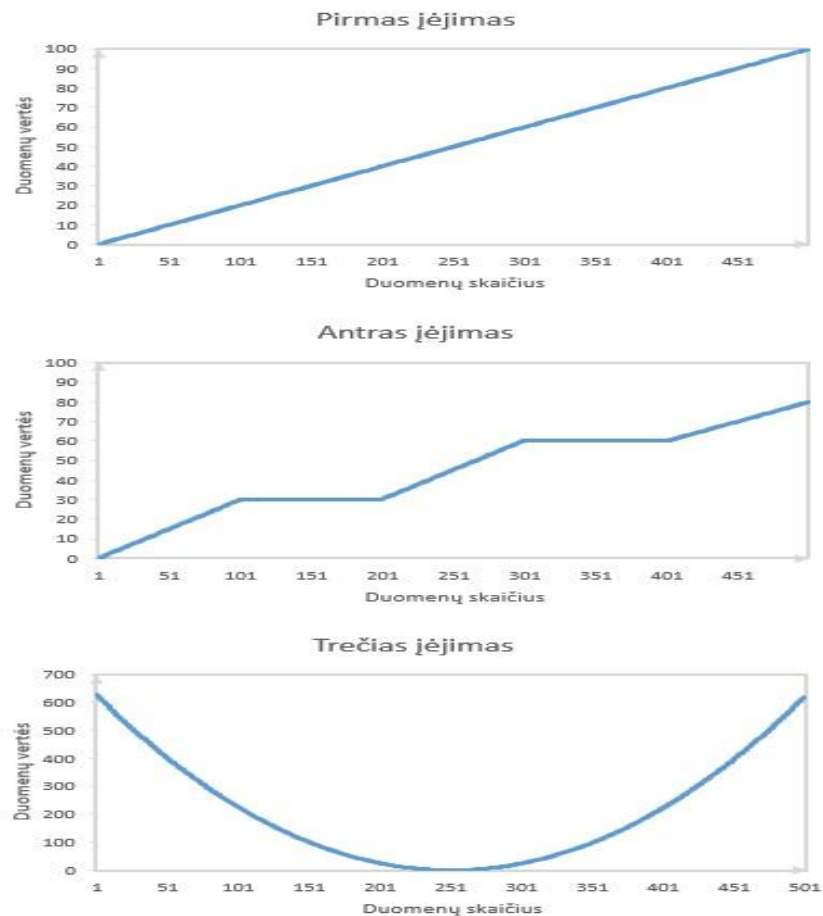
„Duomenys“ srityje yra du mygtukai: „Pridėti“ ir „Išvalyti“. Pirmas mygtukas atidaro Windows Explorer dialogą, kurio pagalba galima surasti norimą duomenų tekstinį failą (.txt), antras, išvalo jau įkeltus duomenis. Įkeliant duomenis būtina, prieš pradėdant konfigūruoti sistemą, įsitikinti, kad jie yra tinkamai suformatuoti, ypač, jei duomenys formatuojami iš Microsoft Excel programinio paketo, kadangi programa atpažins netinkamus failus, tačiau, jei failas tinkamas, bet blogai formatuotas, programa gali nustoti veikti arba neįkelti duomenų. Jau įkeltas išorinių duomenų failas srityje pavaizduotas paveikslėlyje 3.15.



3.15 pav. „Duomenys“ pasirinkimas FT aplinkos posistemės

### 3.4 MATLAB Fuzzy Logic Toolbox ir FuzzyThis rezultatų palyginimas

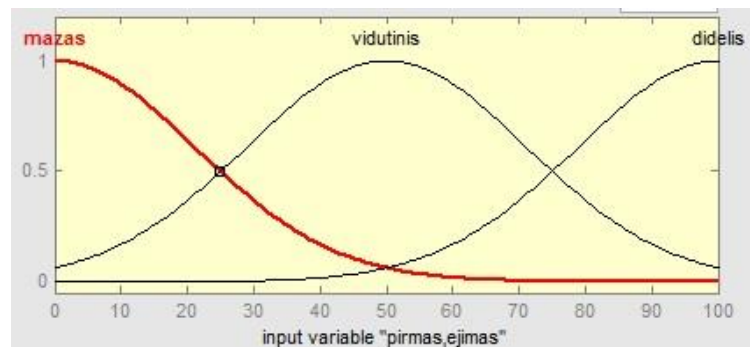
Palyginimui tiek MATLAB Fuzzy Logic Toolbox aplinkoje, tiek naujai sukurtoje FuzzyThis aplinkoje suformuojamos, kuo labiau identiškos sistemos. Duomenys šioms sistemoms sugeneruoti Microsoft Excel paketu. Pirmo įėjimo duomenys nuolat didėja, antro didėja etapais, trečio keičiasi parabolės dėsnio. Visi trys įėjimai pavaizduoti 3.16 paveikslėlyje.



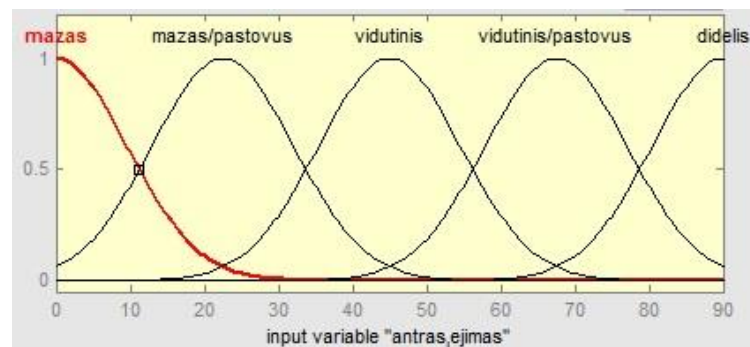
3.16 pav. Sugeneruoti įėjimų duomenys



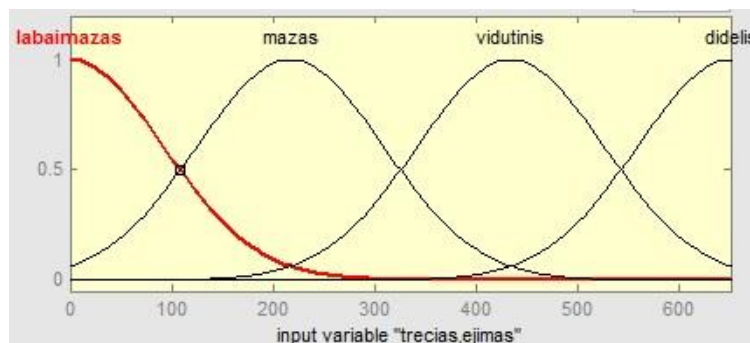
Turint įėjimo duomenis, galima formuoti narystės funkcijas. Naudojau Gauso (Varpo) narystės funkcijų formą, kadangi FuzzyThis paketas šiuo metu palaiko tik šią narystės funkcijos formą. Žemiau esančiuose paveikslėliuose (3.17, 3.18, 3.19) pavaizduotos MATLAB FLT aplinkoje suformuotos narystės funkcijos kiekvienam įėjimui: pirmas įėjimas turi tris narystės funkcijas (mažas, vidutinis, didelis), antras penkias (mažas, mažas/pastovus, vidutinis, vidutinis/pastovus, didelis) ir trečias keturias (labai mažas, mažas, vidutinis, didelis).



**3.17 pav.** Pirmo įėjimo narystės funkcijos MATLAB

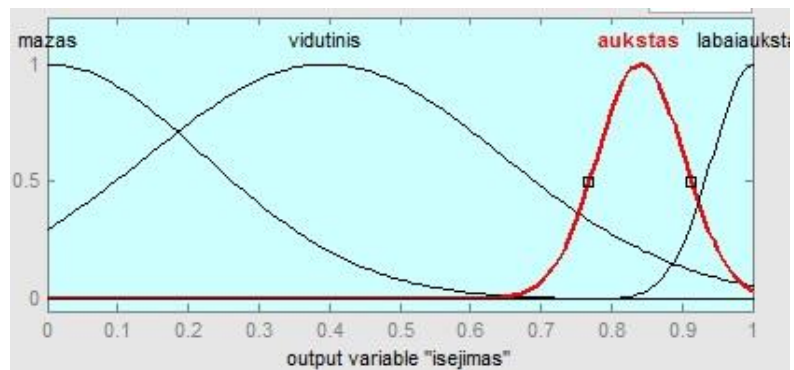


**3.18 pav.** Antro įėjimo narystės funkcijos MATLAB



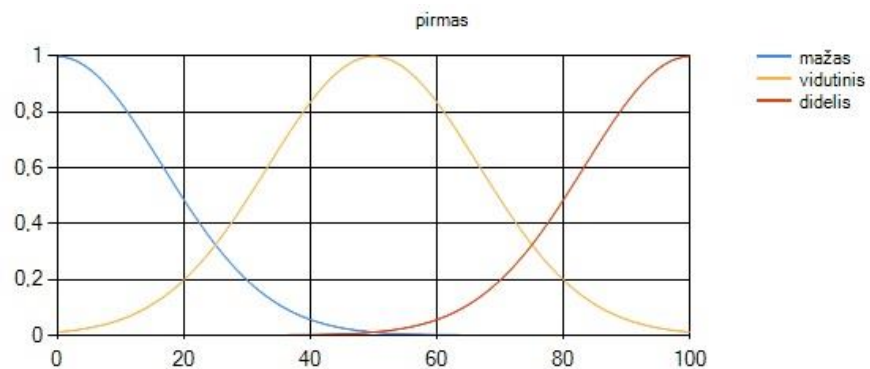
**3.19 pav.** Trečio įėjimo narystės funkcijos MATLAB

Išėjimas turi keturias narystės funkcijas (mažas, vidutinis, aukštas, labai aukštas). Išėjimo narystės funkcijų formos truputį kitokios lyginant su įėjimų narystės funkcijomis, kurių neraiškumai (plotis) buvo maždaug panašus. Šiuo atveju, išėjime labiau apibrėžiamas „aukštas“ ir „labai aukštas“ narystės funkcijos.

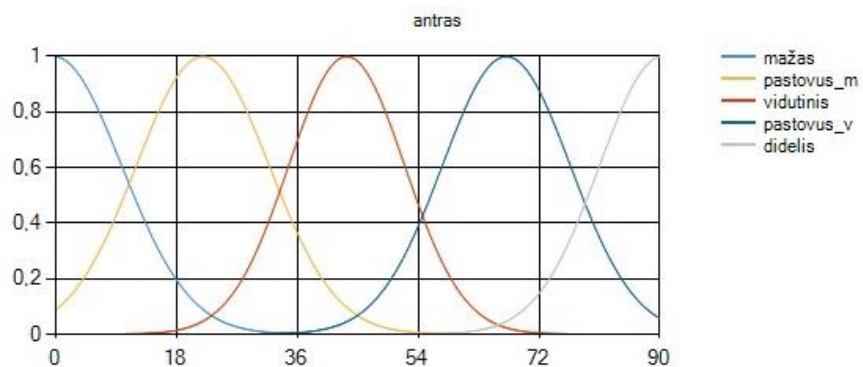


**3.20 pav.** Išėjimo narystės funkcijos MATLAB

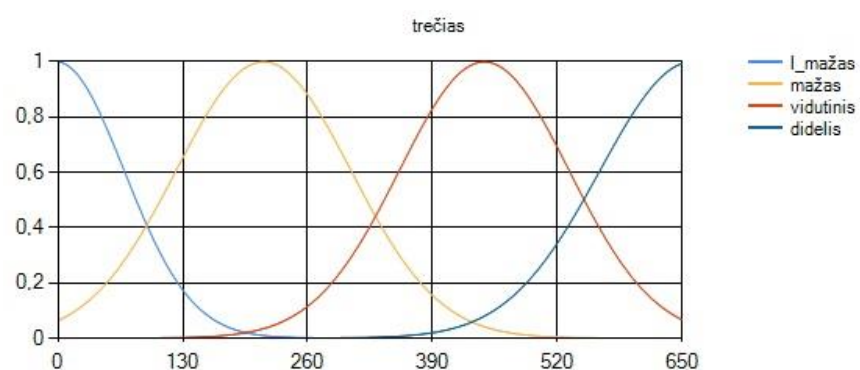
MATLAB FLT aplinkoje suformuoti įėjimai ir išėjimai suformuojami FT aplinkoje.



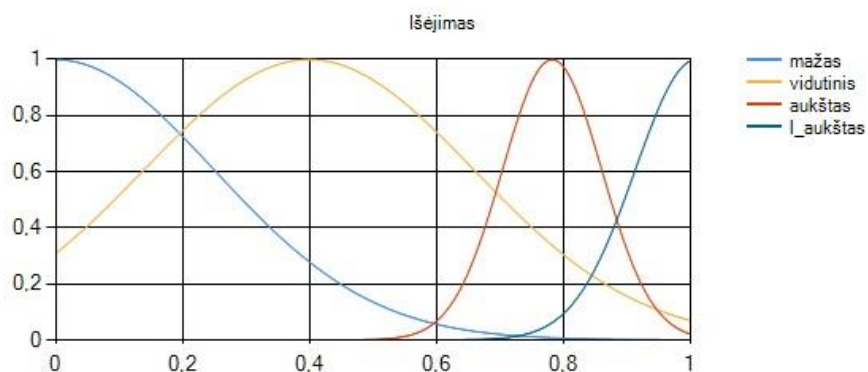
**3.21 pav.** Pirmo įėjimo narystės funkcijos FT



**3.21 pav.** Antro įėjimo narystės funkcijos FT



**3.21 pav.** Trečio įėjimo narystės funkcijos FT



**3.21 pav.** Išėjimo narystės funkcijos FT

Sudarant taisyklės planavau pavaizduoti minimalų rezultatų skirtumą. Tai taisyklių skaičius mažesnis, kad būtų galima lengvai grafiškai nustatyti, koks rezultatas turėtų būti. MFLT aplinkoje sudarytos taisyklės pavaizduotos 3.22 paveikslėlyje, o FT aplinkoje sudarytos taisyklės pavaizduotos 3.23 paveikslėlyje.

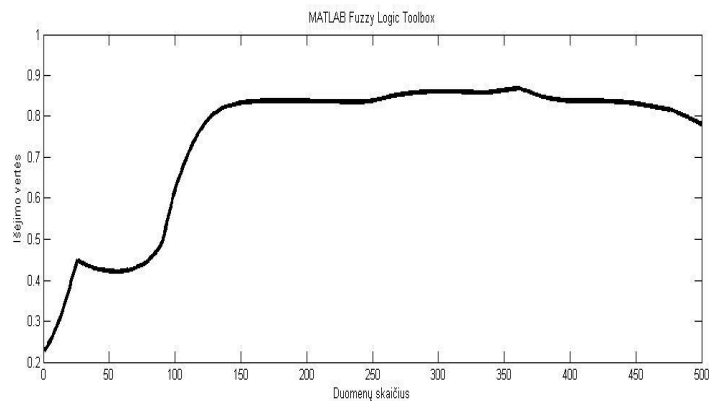
1. If (pirmas\_iejimas is mazas) and (antras\_iejimas is mazas) and (treccias\_iejimas is didelis) then (isejimas is mazas) (1)
2. If (pirmas\_iejimas is mazas) and (antras\_iejimas is mazas/pastovus) and (treccias\_iejimas is vidutinis) then (isejimas is vidutinis) (1)
3. If (pirmas\_iejimas is vidutinis) and (antras\_iejimas is vidutinis) and (treccias\_iejimas is mazas) then (isejimas is aukstas) (1)
4. If (pirmas\_iejimas is didelis) and (antras\_iejimas is vidutinis/pastovus) and (treccias\_iejimas is labaimazas) then (isejimas is labiaukstas) (1)
5. If (pirmas\_iejimas is didelis) and (antras\_iejimas is didelis) and (treccias\_iejimas is labaimazas) then (isejimas is labiaukstas) (1)

### 3.22 pav. Taisyklės MFLT aplinkoje

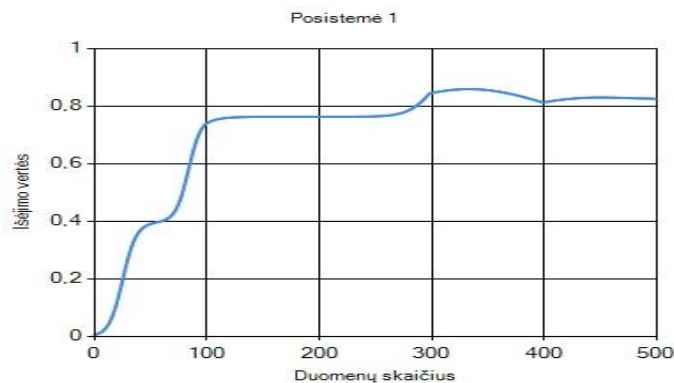
Jei pirmas mažas ir antras mažas ir treccias didelis tai Išėjimas mažas  
 Jei pirmas mažas ir antras pastovus\_m ir treccias vidutinis tai Išėjimas vidutinis  
 Jei pirmas vidutinis ir antras vidutinis ir treccias mažas tai Išėjimas aukštąs  
 Jei pirmas didelis ir antras pastovus\_v ir treccias l\_mažas tai Išėjimas l\_aukštąs  
 Jei pirmas didelis ir antras didelis ir treccias l\_mažas tai Išėjimas l\_aukštąs

### 3.23 pav. Taisyklės FT aplinkoje

Sistemos rezultatai abejose aplinkose pavaizduoti paveikslėliuose 3.24 MFLT ir 3.25 FT. Neturint labai mažai apibrėžtų išėjimo narystės funkcijų ir daug taisyklių rezultatų vertės nesiskiria labai daug. Tačiau ,pavyzdžiui, padidinus „aukštąs“ išėjimo narystės funkcijos plotį (neraiškumą), skirtumas taptų labiau pastebimas.



### 3.24 pav. Taisyklės FT aplinkoje

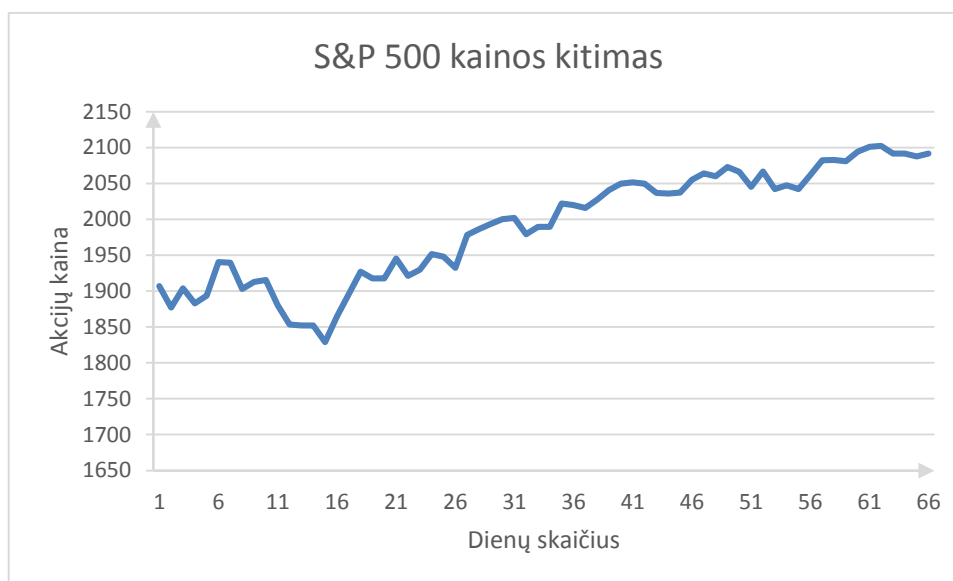


### 3.25 pav. Taisyklės FT aplinkoje

## 4. Taikymo pavyzdžiai

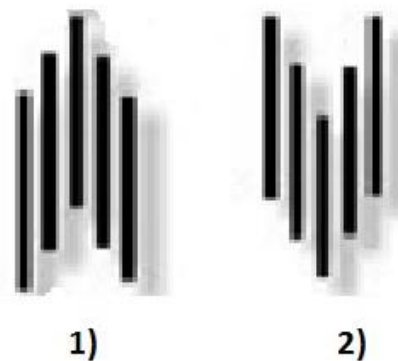
### 4.1 Akcijų kainos augimo/kritimo nustatymas naudojant neraiškių aibių sistemą realizuotą FuzzyThis (FT) aplinkoje

Šio taikymo pavyzdžio tikslas iliustruoti, kad FT programinė aplinka geba tinkamai identifikuoti kilimo kritimo fraktalus/procesus. Duomenys šiai sistemai gaunami iš tinklalapio [www.finance.yahoo.com](http://www.finance.yahoo.com). Tai yra S&P 500 indekso akcijų kaina nuo 2016 sausio 22 iki 2016 balandžio 26, įskaitomos tik darbo dienos. S&P akcijų kaina per 65 dienų minėtą laikotarpį pavaizduota paveikslėlyje 4.1. STANDARD & POOR'S 500 index arba S&P 500 yra vienas pagrindinių indeksų jungtinių Amerikos valstijų (US) akcijų biržoje ir susideda iš 500 didelio kapitalo akcijų, atstovaujamų pirmaujančių US ekonomikos industrijų (energija, medžiagos, finansai, informacinės technologijos ir t.t.). Jos pasirenkamos pagal faktorius kaip rinkos dydis ir nešališkumas arba kitais žodžiais tariant, kaip lengvai turtas ar užstatas gali būti nusipirkta arba parduota nepakeičiant turto kainos.



4.1 pav. S&P 500 indekso kainos kitimas

Sistemos įėjimai formuojami pasinaudojant fraktalais, atsikartojančiais šablonais, kurie gali prognozuoti pasikeitimą didesniame, labiau chaotiškame, kainos kitime. Paprastas fraktalas susideda iš penkių ar daugiau stulpelių. Identifikuojant fraktalą naudojamos taisyklės pavaizduotos 4.2 paveikslėlyje.

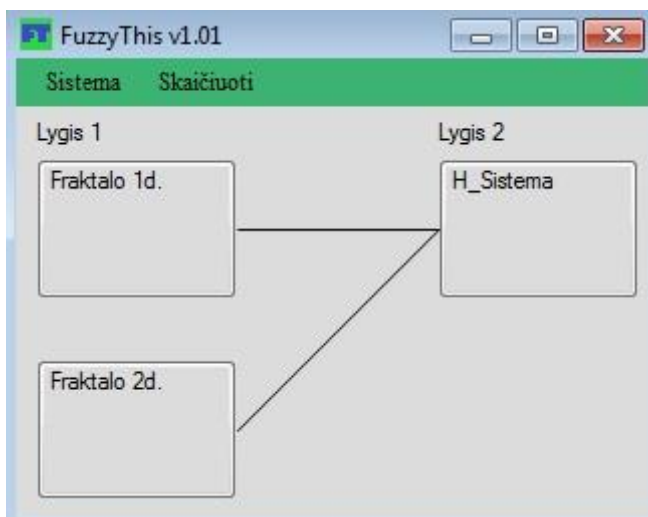


**4.2 pav.** Tobulo fraktalo struktūra

- 1) Idealus mažėjimą nustatantis fraktalas (Ideal Bearish fractal) – nustatomas tada, kuomet šablone didžiausia vertė yra viduriniame stulpelyje ir mažesnės vertės stulpeliuose kiekvienoje pusėje.
- 2) Idealus didėjimą nustatantis fraktalas (Ideal Bullish Fractal) – nustatomas tada, kuomet šablone mažiausia vertė yra viduriniame stulpelyje ir didesnės vertės stulpeliuose kiekvienoje pusėje.

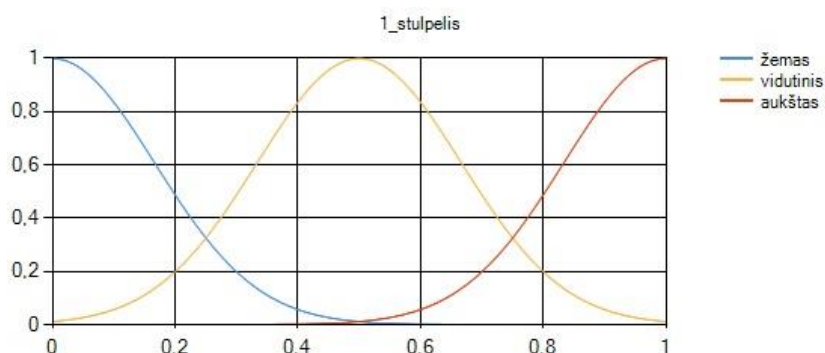
Akivaizdus fraktalų trūkumas yra atsilikimo indikatorius, fraktalas negali būti atpažintas nežinant mažiausiai dviejų pirmų stulpelių verčių, pavyzdžiui, tokiu atveju jei duomenys gaunami kiekvieną dieną – tai fraktalas bus atpažintas tik trečią dieną.

Taigi sistema bus sudaryta iš trijų posistemų (Fraktalo 1d., Fraktalo 2d., H\_Sistema), sujungtų į hierarchinę struktūrą, pavaizduotą 4.3 paveikslėlyje. Pirmoji posistemė apdoros pirmų trijų dienų duomenis t.y. pirmą fraktalo pusę, o antra posistemė apdoros 3, 4, 5 dienų duomenis ir atitiks antrą fraktalo dalį. Akcijos kaina sistemoje keisis penkių dienų intervalu t.y. 1, 2, 3, 4, 5 dienų kaina, kitoje iteracijoje 2, 3, 4, 5, 6 dienų kaina. Tokiu atveju išvadą dėl fraktalo formos bus galima daryti tik penktą dieną nuo sistemos pradžios, o toliau sistema seks kainos kitimą kiekvieną dieną. Fraktalo forma nustatoma H\_Sistema posistemėje apjungus žemesnio lygio posistemų išvadas. Prieš įėjimų duomenis pateikiant sistemai, jie normalizuojami kas penkios dienos, atimant iš duomenų mažiausią vertę ir padalinant iš didžiausios jau atimtos vertės, kad kistų nuo 0 iki 1.

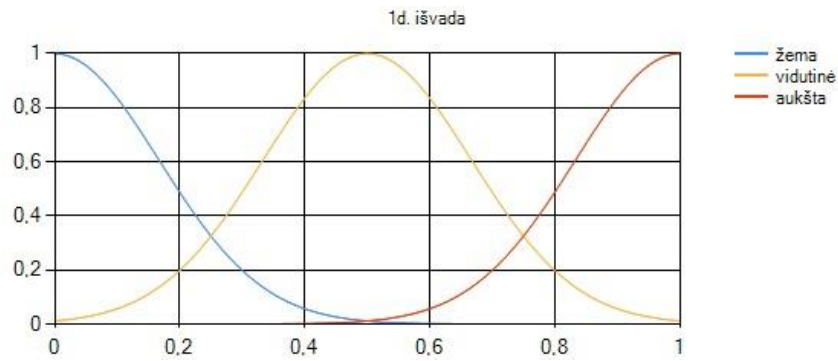


**4.3 pav.** Sistemos sandara FT aplinkoje

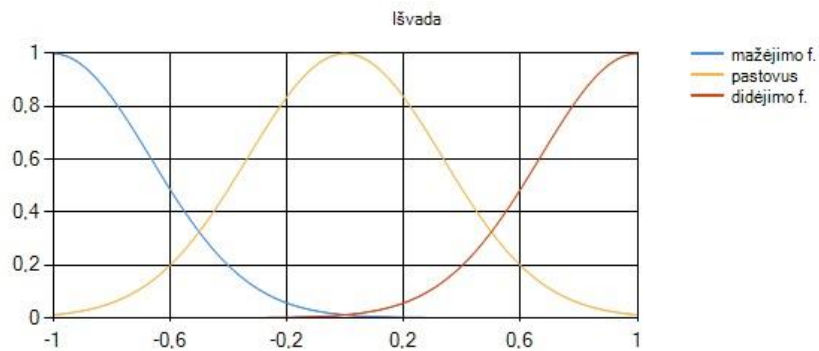
Pirmo lygio posistemės turi po tris įėjimus, visi įėjimai, šiuo atveju, aprašomi vienodai ir jų narystės funkcijos suformuojamos tokios pat formos ir pločio. Suformuotos „Fraktalo 1d.“ posistemės pirmo įėjimo narystės funkcijos pavaizduotos 4.4 paveikslėlyje. „Fraktalo 1d.“ posistemės išėjimo narystės funkcijos pavaizduotos 4.5 paveikslėlyje. „Fraktalo 2d.“ posistemės išėjimo narystės funkcijos, šiuo atveju, yra identiškos „Fraktalo 1d.“ posistemės išėjimo narystės funkcijoms. „H\_Sistema“ posistemė turi du įėjimus, t.y. „Fraktalo 1d.“ ir „Fraktalo 2d.“ posistemų išėjimai. Šių įėjimų narystės funkcijos suformuojamos taip pat kaip ir pirmo lygio posistemų įėjimų narystės funkcijos. Išėjimo narystės funkcijos „H\_Sistema“ posistemiai formuojamos taip pat kaip ir pirmo lygio posistemėms, tačiau dėl norimų išvadų, išėjimo diapazonas parenkamas nuo -1 iki 1, taip visų šio išėjimo narystės funkcijų pločiai (neraiškumai) padvigubėja, o vidurkiai persislenka, kaip pavaizduota 4.6 paveikslėlyje.



**4.4 pav.** „Fraktalo 1d.“ posistemės pirmo įėjimo narystės funkcijos



**4.5 pav.** „Fraktalo 1d.“ posistemės išėjimo narystės funkcijos



**4.6 pav.** „H\_Sistema“ posistemės išėjimo narystės funkcijos

Taisyklės pirmo lygio posistemės, šiuo atveju, yra identiškos. Todėl žemiau pateikiu tik „Fraktalo 1d.“ posistemės įvestas taisykles. Šiose taisyklėse aprašoma pirmą pusę fraktalo formos, kadangi naudojau tik tris narystės funkcijas aprašyti įėjimams, įvertinau ir mažesnes reikšmes aptinkant didėjimą bei didesnes reikšmes aptinkant mažėjimą. „Fraktalo 2d.“ posistemės įvestos taisyklės aprašo antrą fraktalo dalies formą ir galiausiai „H\_Sistema“ posistemės taisyklėse daroma išvada, ar aptinkamas fraktalas, ar ne, ar tai didėjimo fraktalas, ar mažėjimo? „H\_Sistema“ posistemės taisyklės pateikiamos žemiau „Fraktalo 1d.“ taisyklių.

„Fraktalo 1d.“ taisyklės:

- Jei 1 Stulpelis žemas ir 2 Stulpelis vidutinis ir 3 Stulpelis aukštas tai 1d. išvada didėja
- Jei 1 Stulpelis vidutinis ir 2 Stulpelis vidutinis ir 3 Stulpelis aukštas tai 1d. išvada didėja
- Jei 1 Stulpelis žemas ir 2 Stulpelis žemas ir 3 Stulpelis aukštas tai 1d. išvada didėja
- Jei 1 Stulpelis aukštas ir 2 Stulpelis vidutinis ir 3 Stulpelis žemas tai 1d. išvada mažėja
- Jei 1 Stulpelis vidutinis ir 2 Stulpelis vidutinis ir 3 Stulpelis žemas tai 1d. išvada mažėja
- Jei 1 Stulpelis aukštas ir 2 Stulpelis aukštas ir 3 Stulpelis žemas tai 1d. išvada mažėja
- Jei 1 Stulpelis vidutinis ir 2 Stulpelis vidutinis ir 3 Stulpelis vidutinis tai 1d. išvada pastovi
- Jei 1 Stulpelis žemas ir 2 Stulpelis žemas ir 3 Stulpelis žemas tai 1d. išvada pastovi



Jei 1 Stulpelis aukštas ir 2 Stulpelis aukštas ir 3 Stulpelis aukštas tai 1d. išvada pastovi

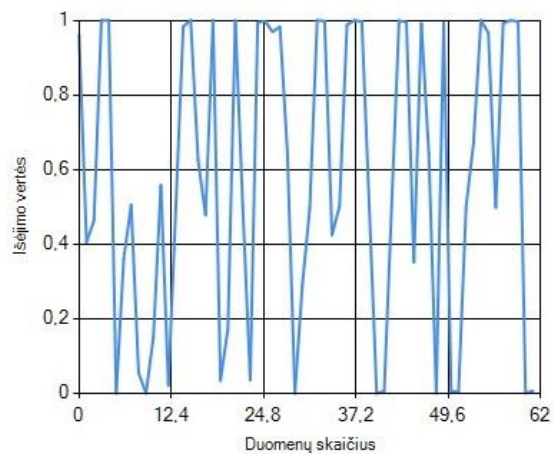
„H\_Sistema“ taisyklės:

Jei Fraktalo 1d. mažėja ir Fraktalo 2d. didėja tai Išvada didėjimo f.

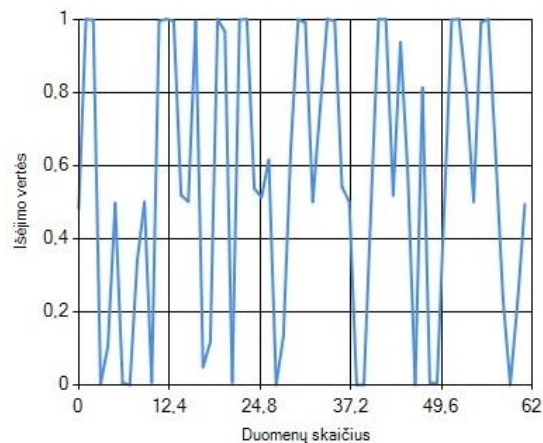
Jei Fraktalo 1d. pastovus ir Fraktalo 2d. pastovus tai Išvada pastovus

Jei Fraktalo 1d. didėja ir Fraktalo 2d. mažėja tai Išvada mažėjimo f.

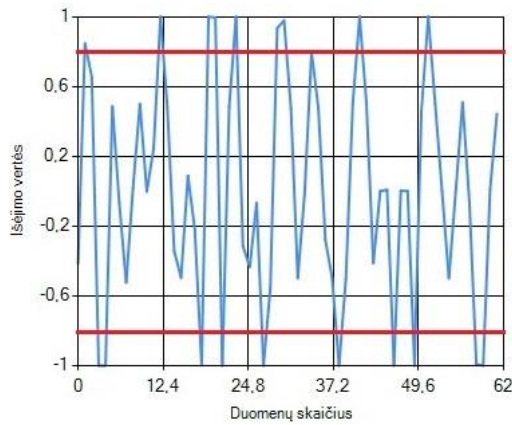
FT programinio paketo apskaičiuotos išėjimų vertės pavaizduotos paveikslėliuose: 4.7 pav. „Fraktalo 1d.“ posistemės išėjimo vertės, 4.8 pav. „Fraktalo 2d.“ posistemės išėjimo vertės ir 4.9 pav. „H\_Sistema“ posistemės išėjimo vertės.



**4.7 pav.** „Fraktalo 1d.“ posistemės išėjimo vertės



**4.8 pav.** „Fraktalo 2d.“ posistemės išėjimo vertės

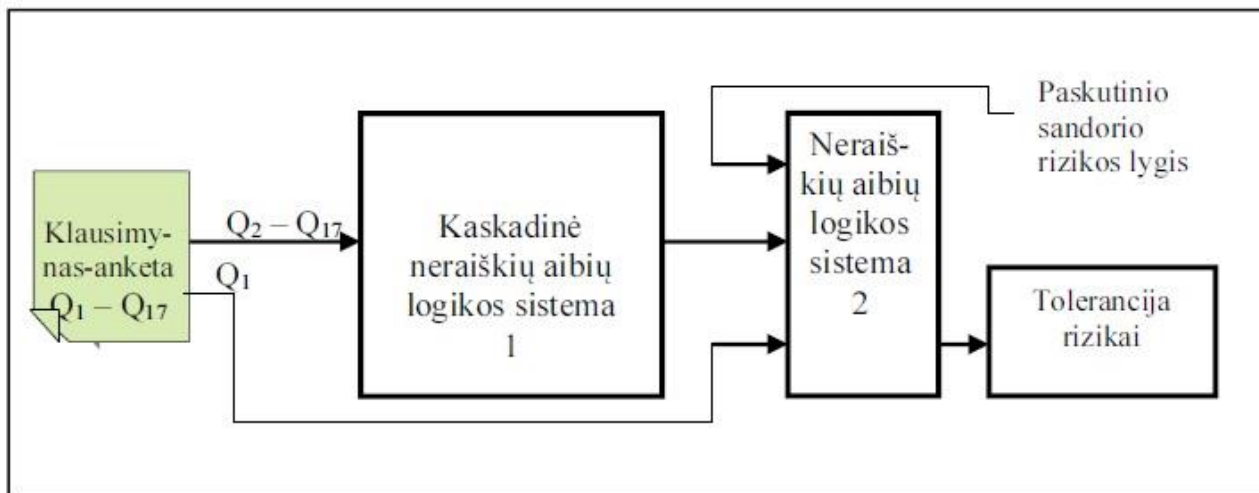


**4.9 pav.** „H\_Sistema“ posistemės išėjimo vertės

4.9 pav. pavaizduota, kad „H\_Sistema“ posistemės išėjimo vertėms pritaikomos ribos (linijos). Kuomet vertės didesnės už 0,8 vertės ribą – vertinama, kad aptinkamas didėjimą nustatantis fraktalas, kai mažesnė už -0,8 vertės ribą – mažėjimą aptinkantis fraktalas. Visos reikšmės esančios tarp šių ribų laikomos pastoviomis ir fraktalo formos nesudarančiomis. Keičiant taisykles ir narystės funkcijų skaičių ir plotį galima įgyvendinti įvairius variantus: sekti, kuomet reikšmė mažėja, kada didėja, ar yra pastovi.

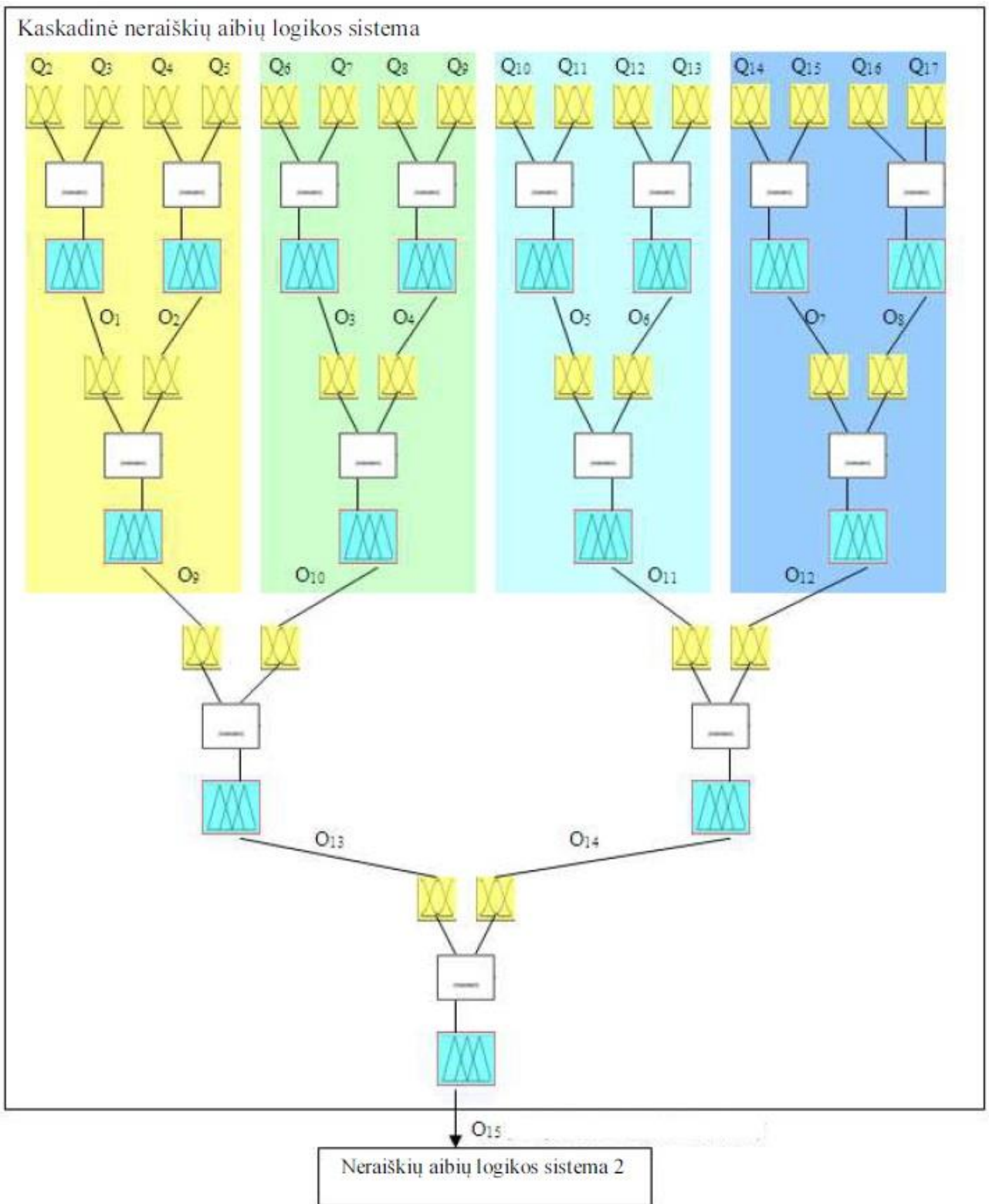
Taigi iš gautų rezultatų galima daryti išvadą, jog Ft aplinkoje sudaryta hierarchinė sistema geba tinkamai identifikuoti kilimo kritimo fraktalus/procesus.

## 4.2 Ekspertinės investuotojo tolerancijos rizikai įvertinimo sistemos realizavimas FuzzyThis (FT) aplinkoje



4.10 pav. Investuotojo tolerancijos rizikai įvertinimo sistemos struktūra

Šio taikymo pavyzdžio tikslas pavaizduoti kaip patogiai ir paprastai FT aplinkoje galima realizuoti komplikuoatą sistemą. A. Jurgučio sudaryta ekspertinė investuotojo tolerancijos rizikai įvertinimo sistemos struktūra pateikta paveikslėlyje 4.10. Ją sudaro klausimynas-anketa, kaskadinė neraiškių aibių logikos sistema ir dar viena neraiškių aibių logikos sistema. Visos sistemos duomenys surenkami iš klausimyno-anketos (investuotojo registracijos formos), kurią sudaro 16 klausimų ir 2 teiginiai (Q2 .. Q5 – Bendrieji klausimai, Q6 .. Q9 – Hipotetiniai klausimai, Q10 .. Q13 – Patirties klausimai, Q14 .. Q17 – Rizikos klausimai). Sistemos ir anketos bei taisyklių sudarymas plačiau aprašytas A. Jurgučio daktaro disertacijoje „Daugiaagentinės ekspertinės investicijų sprendimų paramos sistemos kūrimas ir tyrimai“. Šią anketą taip pat galima užpildyti internetinėje svetainėje <http://www.sprendimutechnologijos.lt/>. Anketos atsakymai į klausimus ar teiginius gali įgauti vertes nuo 0 iki 100, pavyzdžiui, į klausimą „Svarstydami priimti ar nepriimti investicinį sprendimą, Jūsų sprendimą labiau įtakoja galimi menami nuostoliai ar pelnas?“ atsakymas gali būti 0, kuomet jūsų sprendimą labiau įtakoja tik galimas nuostolis arba 100 - kuomet jūsų sprendimą labiau įtakoja tik galimas pelnas. Visos kitos galimos atsakymo vertės tarp šių verčių yra neraiškios – gali įtakoti tiek galimas nuostolis, tiek galimas pelnas. Kaip matome iš paveikslėlio 4.10 kaskadinę neraiškių aibių logikos sistemą sudaro 15 neraiškių aibių posistemų ir O15 posistemės išvada, kuri nustato investuotojo rizikos toleravimą ir eina į antrą neraiškių aibių sistemą, kur A. Jurgutis panaudoja tris posistemas (Aukštai, vidutinei, žemai) rizikos tolerancijai nustatyti. Visi A. Jurgutis skaičiavimai atliekami MATLAB aplinkoje panaudojant Fuzzy Logic Toolbox ir MATLAB komandinės eilutės funkcijas.



4.11 pav. Kaskadinė neraiškių aibių logikos sistema

Norint įsitikinti, ar sukurtas programinis paketas gali spręsti tokius uždavinius, reikia pirma įvertinti A.Jurgučio metodą ir MATLAB FLT aplinkoje gautus rezultatus su atitinkamais klausimyno/anketos duomenimis. Taigi savo nuožiūra atlikau investuotojo registracijos formą internetinėje svetainėje apteiktoje aukščiau.

Investuotojo registracijos formos/anketos rezultatai:

1. Jūs ruošiatės investuoti. Kokiai grupei investuotojų pagal toleranciją rizikai save priskirtumėte?

25

2. Prašome nurodyti savo amžių.

25

3. Nurodykite savo gaunamas mėnesines pajamas.

2100

### **Bendrieji klausimai**

1. Kiek artimai Jūs susipažinę su finansų rinkomis?

30

2. Svarstydami priimti ar nepriimti investicinį sprendimą, Jūsų sprendimą labiau įtakoja galimi menami nuostoliai ar pelnas?

40

3. Po priimto investicinio sprendimo, kaip Jūs dažniausiai jaučiatės?

50

4. Esate girdėję apie infliaciją ir jos daroma neigiamą poveikį Jūsų perkamajai galiai. Priimdami investicinį sprendimą, kiek Jums tinkamas šis teiginys: „Investuoju, nes noriu išlaikyti savo perkamąją galią, nei siekiu kapitalo prieaugio“?

50

### **Hipotetiniai klausimai**

1. Įsivaizduokite, kad mylimiausias giminaitis Jums paliko 100 000 Lt palikimą, nurodęs testamente, kad Jūs visą palikimą privalote investuoti. Kokią dalį palikimo investuotumėte į rizikingas finansines priemones?

25

2. Kaip Jus apibūdintu jūsų geriausias draugas (-ė) rizikos aspektu?

15

3. Tarkime paskutinius 7 metus Jūsų investicijos (kapitalas) nuolat augo. Pastaraisiais metais portfelio vertė nukrito 20 %. Kokią kapitalo (portfelio) dalį keistumėte kitomis investicijomis?

30

4. Tarkime planuojate ateityje imti paskolą. Yra galimybė paskolą gauti su kintančiomis ir (arba) fiksuotomis palūkanomis, o taip pat galimybė paskolą išskaidyti dalimis, kurių viena būtų su

kintančiomis, o kita su fiksuotomis palūkanomis. Kokią dalį paskolos imtumėte su kintančiomis palūkanomis?

40

### **Patirties klausimai**

1. Iš ilgametės statistikos yra paskaičiuota, kad vidutiniškai per metus akcijų rinkose galima uždirbti 12 % gražą. Jūsų nuomone tai sunku ar lengva pasiekti?

70

2. Įvertinkite, kiek Jūs pasitikite savo asmenine patirtimi priimdami, kaip tikitės, pelningą (gerą) investicinį sprendimą?

40

3. Kuria rizikos kryptimi kelis pastaruosius metus keitėsi Jūsų investavimas?

50

4. Ką tik priėmėte investicinį sprendimą, t.y. nusipirkote (arba pardavėte) finansinę priemonę (pvz. akciją, obligaciją ar pan.). Kaip dažniausiai jaučiatės?

50

### **Rizikos klausimai**

1. Investuoto kapitalo vertė laikui bėgant kinta, t.y. kapitalo augimo ir mažėjimo periodai keičia vienas kitą. Kiek investuoto kapitalo vertė gali sumažėti, kad Jūs vis dar jaustumėtės komfortabiliai (išliktumėte ramus)?

25 (sistemoje įvertinama dvigubai didesnė vertė, kadangi anketoje maksimali vertė yra 50)

2. Pasitarę su investicijų konsultantu priimate investicinį sprendimą. Po kurio laiko matote, kad investicinis sprendimas Jums nešą „nekonfortabilų“ nuostolį. Kokią nesėkmės dalį (nuostolio dalį) dėl šio sprendimo prisiimtumėte sau?

70

3. Kiek dažnai priimdami investicinį sprendimą vadovaujatės teiginiu: „Kas nerizikuoja, tas negeria šampano!“ ?

15

4. Kokią riziką toleravote investuodami praeityje? Jei dar nesate investavę, klausimas Jums performuluojamas taip: „Gyvenime Jums tenka patekti į įvairias situacijas, priimti įvairius sprendimus, atlikti įvairius veiksmus. Su koku rizikos lygiu dažniausiai gyvenimiškose situacijose Jums tenka susidurti?“

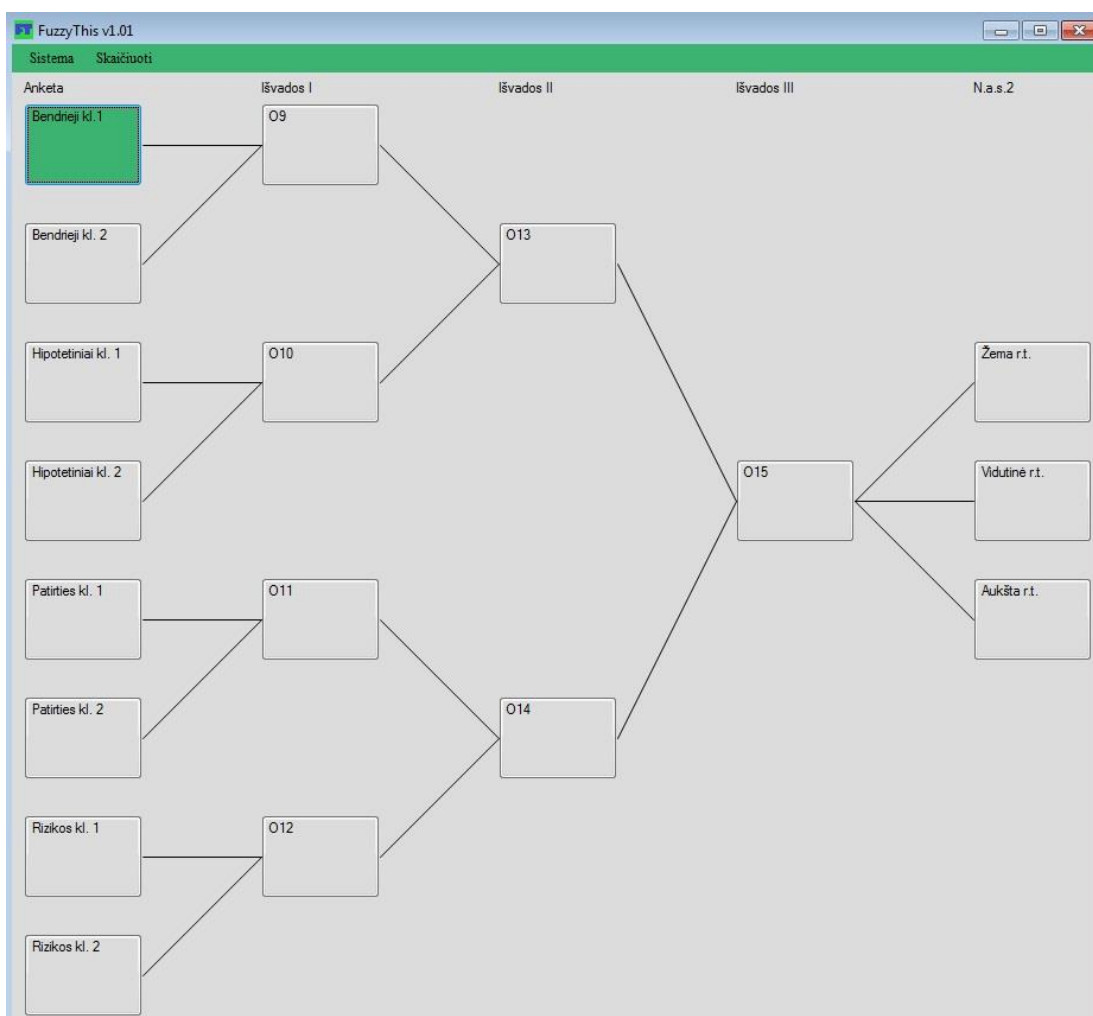
25

Surinktais atliktos anketos duomenimis A.Jurgučio sistema apskaičiuoja rekomenduojamą turto alokaciją ir priskiria prie tolerancijos grupės. Mano įvestais duomenimis, buvau priskirtas žemos rizikos tolerancijos grupei ir rekomenduojama mano turto alokacija:

Rekomenduojama turto alokacija	
Saugios ir likvidžios investicijos (pvz. Indėliai, Vyriausybės obligacijos)	74.97 %
Vidutinės rizikos investicijos (pvz. Įmonių obligacijos, SASO, dividendinės akcijos)	24.33 %
Rizikingos ir spekuliacinės investicijos (pvz. Akcijos, Opcionai, Ateities sandoriai, ETF)	0.71 %

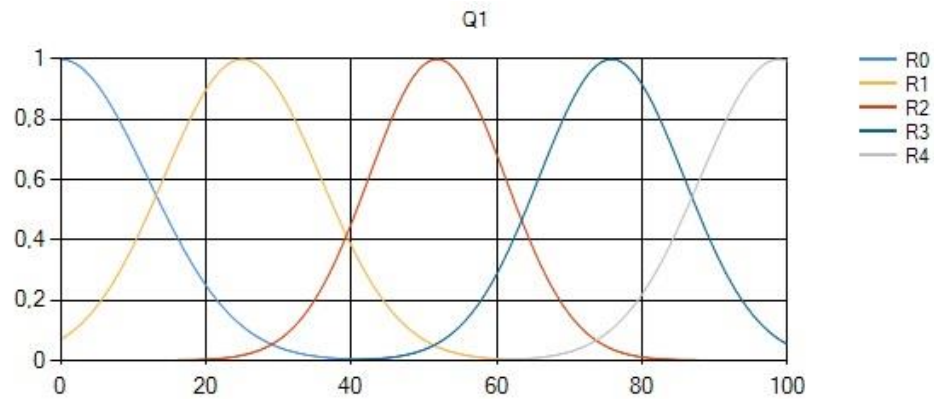
**4.12 pav.** A.Jurgučio sistemos rekomenduojama turto alokacija

Remiantis aukščiau pateiktais anketos duomenimis FT aplinkoje sudaryta visos sistemos hierarchinė neraiškių aibių sistema. Kuri pavaizduota 4.13 pav.

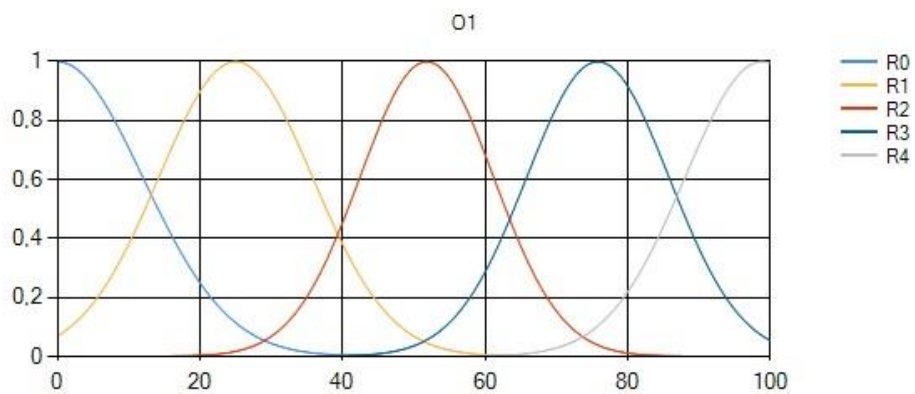


**4.13 pav.** Ekspertinė investuotojo tolerancijos rizikai įvertinimo sistema FT aplinkoje

„Anketa“, „Išvados I“, „Išvados II“, „Išvados III“ lygių posistemės turi po du įėjimus. Šių posistemių, išskyrus O15, įėjimų/išėjimų narystės funkcijos pavaizduotos paveikslėliuose 4.14 ir 4.15. A. Jurgučio sistema ir taisyklės sudarytos remiantis ekspertų duomenimis todėl, šiuo atveju, kaskadinėje neraiškių aibių sistemoje visos posistemės yra identiškos, išskyrus O15.



**4.14 pav.** „Bendrieji kl.1“ posistemės pirmo įėjimo „Q1“ narystės funkcijos.



**4.15 pav.** „Bendrieji kl.1“ posistemės išėjimo „O1“ narystės funkcijos.

Ekspertų išvadų, A.Jurgučio koreguotos, kaskadinės neraiškių aibių logikos sistemos taisyklės kiekvienai posistemiai (R0, R1, R2, R3, R4 - aprašo rizikos lygį):

Jei Q1 R0 ir Q2 R0 tai O1 R0

Jei Q1 R0 ir Q2 R1 tai O1 R0

Jei Q1 R0 ir Q2 R2 tai O1 R1

Jei Q1 R0 ir Q2 R3 tai O1 R1

Jei Q1 R0 ir Q2 R4 tai O1 R2

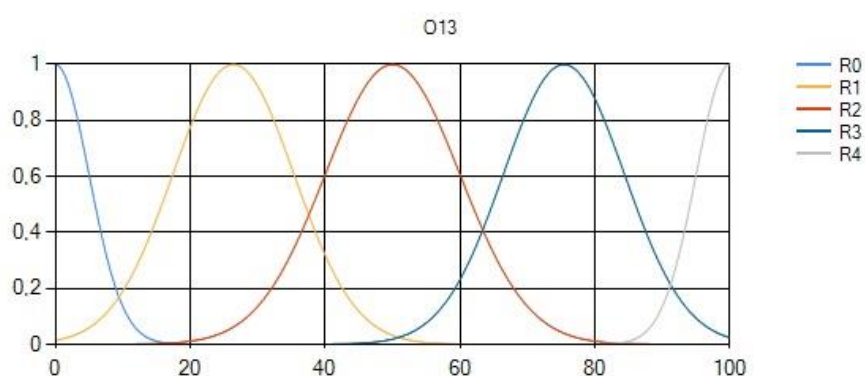
Jei Q1 R1 ir Q2 R0 tai O1 R0

Jei Q1 R1 ir Q2 R1 tai O1 R1

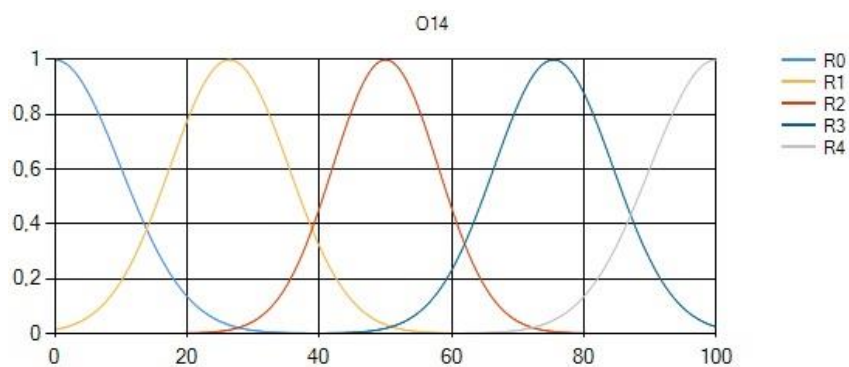


Jei Q1 R1 ir Q2 R2 tai O1 R1  
 Jei Q1 R1 ir Q2 R3 tai O1 R2  
 Jei Q1 R1 ir Q2 R4 tai O1 R2  
 Jei Q1 R2 ir Q2 R0 tai O1 R1  
 Jei Q1 R2 ir Q2 R1 tai O1 R1  
 Jei Q1 R2 ir Q2 R2 tai O1 R2  
 Jei Q1 R2 ir Q2 R3 tai O1 R2  
 Jei Q1 R2 ir Q2 R4 tai O1 R3  
 Jei Q1 R3 ir Q2 R0 tai O1 R1  
 Jei Q1 R3 ir Q2 R1 tai O1 R2  
 Jei Q1 R3 ir Q2 R2 tai O1 R2  
 Jei Q1 R3 ir Q2 R3 tai O1 R3  
 Jei Q1 R3 ir Q2 R4 tai O1 R3  
 Jei Q1 R4 ir Q2 R0 tai O1 R2  
 Jei Q1 R4 ir Q2 R1 tai O1 R2  
 Jei Q1 R4 ir Q2 R2 tai O1 R3  
 Jei Q1 R4 ir Q2 R3 tai O1 R3  
 Jei Q1 R4 ir Q2 R4 tai O1 R4

Kaip minėjau prieš tai, visos kaskadinės neraiškių aibių sistemos posistemų įėjimų/išėjimų narystės funkcijos ir taisyklės yra vienodos, tačiau O15 posistemėje įėjimai truputį pakoreguoti. O15 posistemės įėjimai pateikti 4.16 ir 4.17 pav.

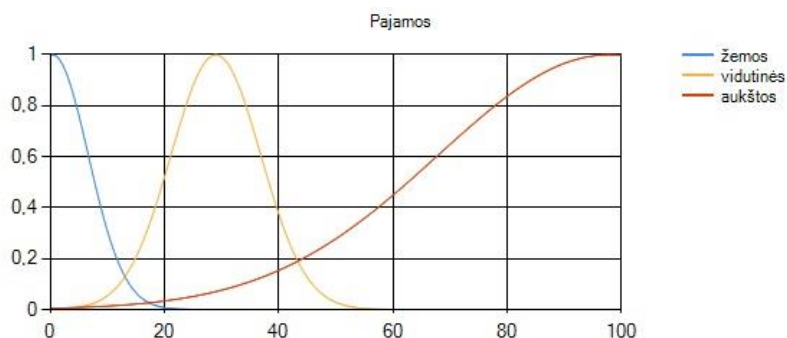


**4.16 pav.** O15 posistemės įėjimo „O13“ narystės funkcijos

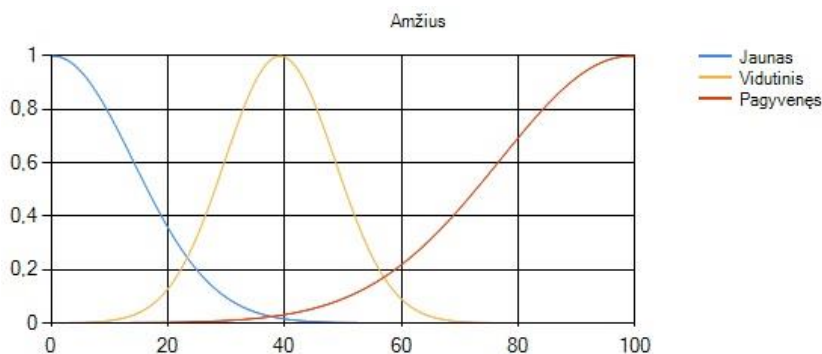


**4.17 pav.** O15 posistemės įėjimo „O14“ narystės funkcijos

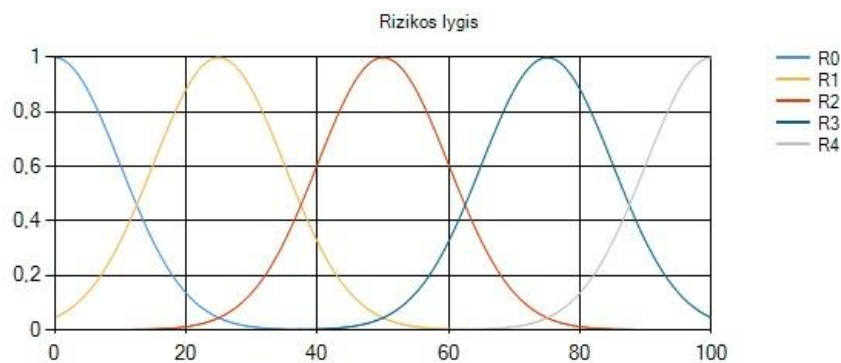
Apskaičiuota O15 posistemės išėjimo vertė lygi 23,6. Turint kaskadinės neraiškių aibių sistemos išvadą, galima skaičiuoti turto alokaciją pasinaudojant antra neraiškių aibių sistema sudaryta iš trijų posistemų (Žema r.t., Vidutinė r.t., Aukšta r.t.). Šios posistemės turės po tris įėjimus: pajamos, amžius ir O15 išėjimas (rizikos lygis). Šių posistemų įėjimų/išėjimų narystės funkcijos, šiuo atveju, yra identiškos, skiriasi tik taisyklės. Įėjimai ir išėjimas pavaizduoti paveikslėliuose 4.18, 4.19, 4.20, 4.21.



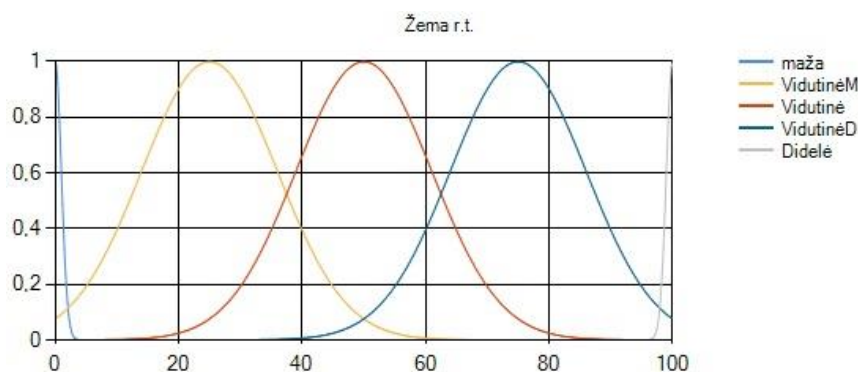
**4.18 pav.** Posistemų įėjimo „Pajamos“ narystės funkcijos



**4.19 pav.** Posistemų įėjimo „Amžius“ narystės funkcijos



4.20 pav. Posistemių įėjimo „Rizikos lygis“ (O15 išėjimas) narystės funkcijos



4.21 pav. „Žema r.t.“ posistemės išėjimo narystės funkcijos

„Žema r.t.“ taisyklės:

- Jei Pajamos žemos ir Amžius Jaunas ir Rizikos lygis R0 tai Saugios inv. Didelė
- Jei Pajamos žemos ir Amžius Jaunas ir Rizikos lygis R1 tai Saugios inv. Didelė
- Jei Pajamos žemos ir Amžius Jaunas ir Rizikos lygis R2 tai Saugios inv. VidutinėD
- Jei Pajamos žemos ir Amžius Jaunas ir Rizikos lygis R3 tai Saugios inv. Vidutinė
- Jei Pajamos žemos ir Amžius Jaunas ir Rizikos lygis R4 tai Saugios inv. Vidutinė
- Jei Pajamos žemos ir Amžius Vidutinis ir Rizikos lygis R0 tai Saugios inv. Didelė
- Jei Pajamos žemos ir Amžius Vidutinis ir Rizikos lygis R1 tai Saugios inv. Didelė
- Jei Pajamos žemos ir Amžius Vidutinis ir Rizikos lygis R2 tai Saugios inv. VidutinėD
- Jei Pajamos žemos ir Amžius Vidutinis ir Rizikos lygis R3 tai Saugios inv. VidutinėD
- Jei Pajamos žemos ir Amžius Vidutinis ir Rizikos lygis R4 tai Saugios inv. Vidutinė
- Jei Pajamos žemos ir Amžius Pagyvenęs ir Rizikos lygis R0 tai Saugios inv. Didelė
- Jei Pajamos žemos ir Amžius Pagyvenęs ir Rizikos lygis R1 tai Saugios inv. Didelė
- Jei Pajamos žemos ir Amžius Pagyvenęs ir Rizikos lygis R2 tai Saugios inv. VidutinėD
- Jei Pajamos žemos ir Amžius Pagyvenęs ir Rizikos lygis R3 tai Saugios inv. VidutinėD

Jei Pajamos žemos ir Amžius Pagyvenęs ir Rizikos lygis R4 tai Saugios inv. VidutinėD  
Jei Pajamos vidutinės ir Amžius Jaunas ir Rizikos lygis R0 tai Saugios inv. Didelė  
Jei Pajamos vidutinės ir Amžius Jaunas ir Rizikos lygis R1 tai Saugios inv. VidutinėD  
Jei Pajamos vidutinės ir Amžius Jaunas ir Rizikos lygis R2 tai Saugios inv. Vidutinė  
Jei Pajamos vidutinės ir Amžius Jaunas ir Rizikos lygis R3 tai Saugios inv. VidutinėM  
Jei Pajamos vidutinės ir Amžius Jaunas ir Rizikos lygis R4 tai Saugios inv. maža  
Jei Pajamos vidutinės ir Amžius Vidutinis ir Rizikos lygis R0 tai Saugios inv. Didelė  
Jei Pajamos vidutinės ir Amžius Vidutinis ir Rizikos lygis R1 tai Saugios inv. Didelė  
Jei Pajamos vidutinės ir Amžius Vidutinis ir Rizikos lygis R2 tai Saugios inv. Vidutinė  
Jei Pajamos vidutinės ir Amžius Vidutinis ir Rizikos lygis R3 tai Saugios inv. Vidutinė  
Jei Pajamos vidutinės ir Amžius Vidutinis ir Rizikos lygis R4 tai Saugios inv. VidutinėM  
Jei Pajamos vidutinės ir Amžius Pagyvenęs ir Rizikos lygis R0 tai Saugios inv. Didelė  
Jei Pajamos vidutinės ir Amžius Pagyvenęs ir Rizikos lygis R1 tai Saugios inv. Didelė  
Jei Pajamos vidutinės ir Amžius Pagyvenęs ir Rizikos lygis R2 tai Saugios inv. VidutinėD  
Jei Pajamos vidutinės ir Amžius Pagyvenęs ir Rizikos lygis R3 tai Saugios inv. Vidutinė  
Jei Pajamos vidutinės ir Amžius Pagyvenęs ir Rizikos lygis R4 tai Saugios inv. Vidutinė  
Jei Pajamos aukštos ir Amžius Jaunas ir Rizikos lygis R0 tai Saugios inv. Didelė  
Jei Pajamos aukštos ir Amžius Jaunas ir Rizikos lygis R1 tai Saugios inv. VidutinėD  
Jei Pajamos aukštos ir Amžius Jaunas ir Rizikos lygis R2 tai Saugios inv. Vidutinė  
Jei Pajamos aukštos ir Amžius Jaunas ir Rizikos lygis R3 tai Saugios inv. maža  
Jei Pajamos aukštos ir Amžius Jaunas ir Rizikos lygis R4 tai Saugios inv. maža  
Jei Pajamos aukštos ir Amžius Vidutinis ir Rizikos lygis R0 tai Saugios inv. Didelė  
Jei Pajamos aukštos ir Amžius Vidutinis ir Rizikos lygis R1 tai Saugios inv. VidutinėD  
Jei Pajamos aukštos ir Amžius Vidutinis ir Rizikos lygis R2 tai Saugios inv. Vidutinė  
Jei Pajamos aukštos ir Amžius Vidutinis ir Rizikos lygis R3 tai Saugios inv. VidutinėM  
Jei Pajamos aukštos ir Amžius Vidutinis ir Rizikos lygis R4 tai Saugios inv. maža  
Jei Pajamos aukštos ir Amžius Pagyvenęs ir Rizikos lygis R0 tai Saugios inv. Didelė  
Jei Pajamos aukštos ir Amžius Pagyvenęs ir Rizikos lygis R1 tai Saugios inv. Didelė  
Jei Pajamos aukštos ir Amžius Pagyvenęs ir Rizikos lygis R2 tai Saugios inv. Vidutinė  
Jei Pajamos aukštos ir Amžius Pagyvenęs ir Rizikos lygis R3 tai Saugios inv. VidutinėM  
Jei Pajamos aukštos ir Amžius Pagyvenęs ir Rizikos lygis R4 tai Saugios inv. Maža  
„Žema r.t.“ posistemės išėjimo vertė lygi 99,85.

„Vidutinė r.t.“ taisyklės:

Jei Pajamos žemos ir Amžius Jaunas ir Rizikos lygis R0 tai Saugios inv. maža  
Jei Pajamos žemos ir Amžius Jaunas ir Rizikos lygis R1 tai Saugios inv. maža  
Jei Pajamos žemos ir Amžius Jaunas ir Rizikos lygis R2 tai Saugios inv. Vidutinė  
Jei Pajamos žemos ir Amžius Jaunas ir Rizikos lygis R3 tai Saugios inv. Vidutinė  
Jei Pajamos žemos ir Amžius Jaunas ir Rizikos lygis R4 tai Saugios inv. Didelė  
Jei Pajamos žemos ir Amžius Vidutinis ir Rizikos lygis R0 tai Saugios inv. maža  
Jei Pajamos žemos ir Amžius Vidutinis ir Rizikos lygis R1 tai Saugios inv. maža  
Jei Pajamos žemos ir Amžius Vidutinis ir Rizikos lygis R2 tai Saugios inv. VidutinėM  
Jei Pajamos žemos ir Amžius Vidutinis ir Rizikos lygis R3 tai Saugios inv. Vidutinė  
Jei Pajamos žemos ir Amžius Vidutinis ir Rizikos lygis R4 tai Saugios inv. VidutinėD  
Jei Pajamos žemos ir Amžius Pagyvenęs ir Rizikos lygis R0 tai Saugios inv. maža  
Jei Pajamos žemos ir Amžius Pagyvenęs ir Rizikos lygis R1 tai Saugios inv. maža  
Jei Pajamos žemos ir Amžius Pagyvenęs ir Rizikos lygis R2 tai Saugios inv. VidutinėM  
Jei Pajamos žemos ir Amžius Pagyvenęs ir Rizikos lygis R3 tai Saugios inv. VidutinėM  
Jei Pajamos žemos ir Amžius Pagyvenęs ir Rizikos lygis R4 tai Saugios inv. Vidutinė  
Jei Pajamos vidutinės ir Amžius Jaunas ir Rizikos lygis R0 tai Saugios inv. maža  
Jei Pajamos vidutinės ir Amžius Jaunas ir Rizikos lygis R1 tai Saugios inv. VidutinėM  
Jei Pajamos vidutinės ir Amžius Jaunas ir Rizikos lygis R2 tai Saugios inv. Didelė  
Jei Pajamos vidutinės ir Amžius Jaunas ir Rizikos lygis R3 tai Saugios inv. Didelė  
Jei Pajamos vidutinės ir Amžius Jaunas ir Rizikos lygis R4 tai Saugios inv. VidutinėD  
Jei Pajamos vidutinės ir Amžius Vidutinis ir Rizikos lygis R0 tai Saugios inv. maža  
Jei Pajamos vidutinės ir Amžius Vidutinis ir Rizikos lygis R1 tai Saugios inv. VidutinėM  
Jei Pajamos vidutinės ir Amžius Vidutinis ir Rizikos lygis R2 tai Saugios inv. VidutinėD  
Jei Pajamos vidutinės ir Amžius Vidutinis ir Rizikos lygis R3 tai Saugios inv. Didelė  
Jei Pajamos vidutinės ir Amžius Vidutinis ir Rizikos lygis R4 tai Saugios inv. Didelė  
Jei Pajamos vidutinės ir Amžius Pagyvenęs ir Rizikos lygis R0 tai Saugios inv. maža  
Jei Pajamos vidutinės ir Amžius Pagyvenęs ir Rizikos lygis R1 tai Saugios inv. maža  
Jei Pajamos vidutinės ir Amžius Pagyvenęs ir Rizikos lygis R2 tai Saugios inv. VidutinėM  
Jei Pajamos vidutinės ir Amžius Pagyvenęs ir Rizikos lygis R3 tai Saugios inv. Vidutinė  
Jei Pajamos vidutinės ir Amžius Pagyvenęs ir Rizikos lygis R4 tai Saugios inv. VidutinėD  
Jei Pajamos aukštos ir Amžius Jaunas ir Rizikos lygis R0 tai Saugios inv. maža  
Jei Pajamos aukštos ir Amžius Jaunas ir Rizikos lygis R1 tai Saugios inv. VidutinėM  
Jei Pajamos aukštos ir Amžius Jaunas ir Rizikos lygis R2 tai Saugios inv. Didelė  
Jei Pajamos aukštos ir Amžius Jaunas ir Rizikos lygis R3 tai Saugios inv. Didelė

Jei Pajamos aukštos ir Amžius Jaunas ir Rizikos lygis R4 tai Saugios inv. VidutinėD  
Jei Pajamos aukštos ir Amžius Vidutinis ir Rizikos lygis R0 tai Saugios inv. maža  
Jei Pajamos aukštos ir Amžius Vidutinis ir Rizikos lygis R1 tai Saugios inv. VidutinėM  
Jei Pajamos aukštos ir Amžius Vidutinis ir Rizikos lygis R2 tai Saugios inv. Didelė  
Jei Pajamos aukštos ir Amžius Vidutinis ir Rizikos lygis R3 tai Saugios inv. Didelė  
Jei Pajamos aukštos ir Amžius Vidutinis ir Rizikos lygis R4 tai Saugios inv. VidutinėD  
Jei Pajamos aukštos ir Amžius Pagyvenęs ir Rizikos lygis R0 tai Saugios inv. maža  
Jei Pajamos aukštos ir Amžius Pagyvenęs ir Rizikos lygis R1 tai Saugios inv. VidutinėM  
Jei Pajamos aukštos ir Amžius Pagyvenęs ir Rizikos lygis R2 tai Saugios inv. Vidutinė  
Jei Pajamos aukštos ir Amžius Pagyvenęs ir Rizikos lygis R3 tai Saugios inv. Didelė  
Jei Pajamos aukštos ir Amžius Pagyvenęs ir Rizikos lygis R4 tai Saugios inv. Didelė  
„Vidutinė r.t.“ posistemės išėjimo vertė lygi 16,36.

„Aukšta r.t.“ taisyklės:

Jei Pajamos žemos ir Amžius Jaunas ir Rizikos lygis R0 tai Saugios inv. maža  
Jei Pajamos žemos ir Amžius Jaunas ir Rizikos lygis R1 tai Saugios inv. maža  
Jei Pajamos žemos ir Amžius Jaunas ir Rizikos lygis R2 tai Saugios inv. maža  
Jei Pajamos žemos ir Amžius Jaunas ir Rizikos lygis R3 tai Saugios inv. VidutinėM  
Jei Pajamos žemos ir Amžius Jaunas ir Rizikos lygis R4 tai Saugios inv. Vidutinė  
Jei Pajamos žemos ir Amžius Vidutinis ir Rizikos lygis R0 tai Saugios inv. maža  
Jei Pajamos žemos ir Amžius Vidutinis ir Rizikos lygis R1 tai Saugios inv. maža  
Jei Pajamos žemos ir Amžius Vidutinis ir Rizikos lygis R2 tai Saugios inv. maža  
Jei Pajamos žemos ir Amžius Vidutinis ir Rizikos lygis R3 tai Saugios inv. VidutinėM  
Jei Pajamos žemos ir Amžius Vidutinis ir Rizikos lygis R4 tai Saugios inv. VidutinėM  
Jei Pajamos žemos ir Amžius Pagyvenęs ir Rizikos lygis R0 tai Saugios inv. maža  
Jei Pajamos žemos ir Amžius Pagyvenęs ir Rizikos lygis R1 tai Saugios inv. maža  
Jei Pajamos žemos ir Amžius Pagyvenęs ir Rizikos lygis R2 tai Saugios inv. maža  
Jei Pajamos žemos ir Amžius Pagyvenęs ir Rizikos lygis R3 tai Saugios inv. maža  
Jei Pajamos žemos ir Amžius Pagyvenęs ir Rizikos lygis R4 tai Saugios inv. maža  
Jei Pajamos vidutinės ir Amžius Jaunas ir Rizikos lygis R0 tai Saugios inv. maža  
Jei Pajamos vidutinės ir Amžius Jaunas ir Rizikos lygis R1 tai Saugios inv. maža  
Jei Pajamos vidutinės ir Amžius Jaunas ir Rizikos lygis R2 tai Saugios inv. VidutinėM  
Jei Pajamos vidutinės ir Amžius Jaunas ir Rizikos lygis R3 tai Saugios inv. VidutinėD  
Jei Pajamos vidutinės ir Amžius Jaunas ir Rizikos lygis R4 tai Saugios inv. Didelė  
Jei Pajamos vidutinės ir Amžius Vidutinis ir Rizikos lygis R0 tai Saugios inv. maža

Jei Pajamos vidutinės ir Amžius Vidutinis ir Rizikos lygis R1 tai Saugios inv. maža  
 Jei Pajamos vidutinės ir Amžius Vidutinis ir Rizikos lygis R2 tai Saugios inv. VidutinėM  
 Jei Pajamos vidutinės ir Amžius Vidutinis ir Rizikos lygis R3 tai Saugios inv. Vidutinė  
 Jei Pajamos vidutinės ir Amžius Vidutinis ir Rizikos lygis R4 tai Saugios inv. VidutinėD  
 Jei Pajamos vidutinės ir Amžius Pagyvenęs ir Rizikos lygis R0 tai Saugios inv. maža  
 Jei Pajamos vidutinės ir Amžius Pagyvenęs ir Rizikos lygis R1 tai Saugios inv. maža  
 Jei Pajamos vidutinės ir Amžius Pagyvenęs ir Rizikos lygis R2 tai Saugios inv. VidutinėM  
 Jei Pajamos vidutinės ir Amžius Pagyvenęs ir Rizikos lygis R3 tai Saugios inv. Vidutinė  
 Jei Pajamos vidutinės ir Amžius Pagyvenęs ir Rizikos lygis R4 tai Saugios inv. Vidutinė  
 Jei Pajamos aukštos ir Amžius Jaunas ir Rizikos lygis R0 tai Saugios inv. maža  
 Jei Pajamos aukštos ir Amžius Jaunas ir Rizikos lygis R1 tai Saugios inv. VidutinėM  
 Jei Pajamos aukštos ir Amžius Jaunas ir Rizikos lygis R2 tai Saugios inv. Vidutinė  
 Jei Pajamos aukštos ir Amžius Jaunas ir Rizikos lygis R3 tai Saugios inv. VidutinėD  
 Jei Pajamos aukštos ir Amžius Jaunas ir Rizikos lygis R4 tai Saugios inv. Didelė  
 Jei Pajamos aukštos ir Amžius Vidutinis ir Rizikos lygis R0 tai Saugios inv. maža  
 Jei Pajamos aukštos ir Amžius Vidutinis ir Rizikos lygis R1 tai Saugios inv. VidutinėM  
 Jei Pajamos aukštos ir Amžius Vidutinis ir Rizikos lygis R2 tai Saugios inv. Vidutinė  
 Jei Pajamos aukštos ir Amžius Vidutinis ir Rizikos lygis R3 tai Saugios inv. VidutinėD  
 Jei Pajamos aukštos ir Amžius Vidutinis ir Rizikos lygis R4 tai Saugios inv. Didelė  
 Jei Pajamos aukštos ir Amžius Pagyvenęs ir Rizikos lygis R0 tai Saugios inv. maža  
 Jei Pajamos aukštos ir Amžius Pagyvenęs ir Rizikos lygis R1 tai Saugios inv. maža  
 Jei Pajamos aukštos ir Amžius Pagyvenęs ir Rizikos lygis R2 tai Saugios inv. Vidutinė  
 Jei Pajamos aukštos ir Amžius Pagyvenęs ir Rizikos lygis R3 tai Saugios inv. Vidutinė  
 Jei Pajamos aukštos ir Amžius Pagyvenęs ir Rizikos lygis R4 tai Saugios inv. VidutinėD  
 „Aukšta r.t.“ posistemės išėjimo vertė lygi 0,74.

Taigi šių trijų posistemų rezultatai yra: žema r.t.(99,85), vidutinė r.t.(16,36), aukšta r.t. (0.74).

Apskaičiuojamas rekomenduojamas procentinis turto paskirstymas:

$$Suma = 99,85 + 16,36 + 0,74 = 116,95$$

$$Saugios\ ir\ likvidžios\ investicijos = \left( \frac{99,85}{116,95} \right) * 100 = 85,38 \%$$

$$Vidutinės\ rizikos\ investicijos = \left( \frac{16,36}{116,95} \right) * 100 = 13,99 \%$$

$$Rizikingos\ ir\ spekuliacinės\ investicijos = \left( \frac{0,74}{116,95} \right) * 100 = 0,63 \%$$

### 1. lentelė Rekomenduojama turto alokacija

Rekomenduojama turto alokacija	MFLT	FT
Saugios ir likvidžios investicijos	74,97 %	85,38 %
Vidutinės rizikos investicijos	24,33 %	13,99 %
Rizikingos ir spekuliacinės investicijos	0,71 %	0,63 %

Kaip matome iš 1. Lentelės pateiktų rezultatų, FT sukurtas programinis paketas mano anketos duomenis taip pat priskiria prie žemos rizikos tolerancijos. Rezultatai skiriasi, esu „labiau“ priskiriamas su FT aplinka. Šis skirtumas susidaro dėl to, kad A.Jurgutis MFLT aplinkoje naudojo trikampes ir trapecines narystės funkcijas, kuomet aš FT aplinkoje naudoju gausines narystės funkcijas. Dar vienas aspektas įtakojąs skirtumą yra narystės funkcijos neraiškumas (plotis). 4.21 pav. matoma, kad narystės funkcijos „maža“ ir „Didelė“ turi labai mažą plotį (neraiškumą), todėl FT „įvertindama neraiškumą, skaičiuoja vertes labiau atsižvelgdama į šių narystės funkcijų vidurkius.

Taigi komplikauta A.Jurgučio sistema buvo paprastai ir greitai realizuota FT aplinkoje ir gauti panašūs rezultatai demonstruoja pasiūlyto programinio paketo darbingumą/veiksmingumą.



## 5. Išvados

1. Neraiškių aibių logika leidžia patogiai formalizuoti procesų eksperų žinias ir sprendimus, tačiau šiuo metu programiniuose paketuose neraiškumo eliminavimui dažniausiai naudojamas apjungtų taisyklių išėjimo kintamojo priklausomybės svorio centro nustatymo metodas, kuris nevertina išėjimo kintamojo apibrėžtumo.
2. Analizė parodė, kad egzistuojantys neraiškių aibių logikos programiniai paketai turi savo pranašumų ir trūkumų. Vieni nėra autonomiški, kiti brangūs, dar kiti reikalauja daugiau priežiūros. Tačiau dauguma analizuotų paketų defuzifikavimo procedūrai naudoja apjungtų taisyklių svorio centro principą.
3. Sukurto programinio paketo įėjimų/išėjimų funkcinės priklausomybės leidžia įvertinti sistemos narystės funkcijų neraiškumus, kadangi defuzifikavimo procedūroje narystės funkcijos traktuojamos, kaip tikimybinių skirstinių aproksimacija.
4. Paketo programavimo terpė yra Microsoft Visual Studio 2015, o programavimo kalba - Visual C#, kuri palaiko .NET struktūrą. Todėl sukurtas programinis paketas gali būti lengvai integruotas į programinę įrangą veikiančią su .NET struktūra.
5. Sukurtas programinis paketas FuzzyThis turi visas reikalingas funkcijas kurti neraiškių aibių sistemas ir galimybę kurti hierarchinės struktūros neraiškių aibių sistemas.
6. Pasiūlyta neraiškių aibių sistema ypač tinka tokiems atvejams, kada įvairių taisyklių išėjimo kintamųjų priklausomybės funkcijos turi skirtingus pločius (t.y. labai skiriasi taisyklių taikymo neapibrėžtumas). Kada priklausomybės funkcijų pločiai yra panašūs, gaunami sistemos išėjimai yra panašūs, kaip ir taikant tradicines neraiškių aibių sistemas.
7. Pasiūlytos įrangos testavimas parodė, kad FT aplinkoje sudaryta hierarchinė neraiškių aibių sistema tinkamai identifikuoja S&P 500 indekso kainos kitimo mažėjimo ir didėjimo fraktalus.
8. Programos testai taip pat parodė, kad FT aplinkoje apskaičiuoti hierarchinės ekspertinės investuotojo rizikos sistemos rezultatai yra panašūs su kitų autorių gautais rezultatais, todėl pasiūlytą sistemą, kaip autonominę programinę įrangą, galima sėkmingai taikyti ir kituose hierarchinėse neraiškių aibių sistemose.

## 6. Literatūra

1. R. Simutis, V. Galvanauskas. *Hibridinės procesų stebėsenos ir valdymo sistemos (Medžiaga egzaminui)*. Kaunas: Kauno Technologijos Universitetas. [34 -35 psl.]
2. R. Simutis, V. Galvanauskas. *Hibridinės procesų stebėsenos ir valdymo sistemos (Medžiaga koliokviumui)*. Kaunas: Kauno Technologijos Universitetas. [30-46 psl.]
3. Yong – Zai Lu. *Industrial intelligent control* [25 -76 psl.].
4. M. Negnevitsky. *Artificial intelligence*. [106-148 psl.]
5. Valluru B. Rao. *C++ Neural Networks and Fuzzy Logic*. [skyriai: 3,9,16]
6. Michael J. Pont. *Software Engineering with C++ and Case Tools*. 1996
7. Snehal N. Rajguru\*, Pritesh Shah, NeelaRayavarapu and PriyankaTupe. *Application of fuzzy controllers for flow control processes in chemical industries, 2014*.
8. E Kolman; M Margaliot. *Knowledge-Based Neurocomputing: A Fuzzy Logic Approach, Springer Science & Business Media Publications, 2009*.
9. Internetinė prieiga: <https://msdn.microsoft.com/en-us/library/>
10. Internetinė prieiga: <http://www.homeandlearn.co.uk/csharp/csharp.html/>
11. J. Sharp, J Jagger „*Microsoft visual c#.net step by step*“, 2002
12. Internetinė prieiga: [www.finance.yahoo.com](http://www.finance.yahoo.com)
13. Internetinė prieiga: <http://www.sprendimutechnologijos.lt/>
14. Wang, Shuming. *Fuzzy random optimization : theory, models, and applications*. 2012
15. Internetinė prieiga: <http://www.fuzzytech.com/>
16. R. A. Orchard. *FuzzyCLIPS Version 6.04A User's Guide*. 1998
17. HyperLogic TECHNICAL NOTES. *Fuzzy Logic Inference Techniques in CubiCalc Version 2*. Šaltinis: [www.hyperlogic.com/](http://www.hyperlogic.com/)
18. Internetinė prieiga: <http://se.mathworks.com/products/fuzzy-logic/>
19. A. Jurgutis. *Daugiaagentinės ekspertinės investicinių sprendimų paramos sistemos kūrimas ir tyrimai*. [101-106 psl.]
20. J.Kuepper. *A Trader's Guide To Using Fractals*.

## 7. Priedai

Prieduose nepateikiamas Microsoft Visual Studio sugeneruotas kodas.

```
Funkcijos.cs
using System;
using System.Linq;

namespace FT_Final_v1._0_Smart
{
    class Funkcijos
    {
        public static void masyvo_elemento_istrinimas<T>(ref T[] originalas,int indeksas)
        {
            for(int i= indeksas + 1; i < originalas.Length; i++)
            {
                originalas[i - 1] = originalas[i];
            }
            Array.Resize(ref originalas, originalas.Length - 1);
        }
        public static void masyvo_dydzio_keitimas_2d<T>(ref T[,] originalas, int nx, int ny)
        {
            var naujasmasyvas = new T[nx, ny];
            int x = Math.Min(nx, originalas.GetLength(0));
            int y = Math.Min(ny, originalas.GetLength(1));
            for (int i = 0; i < x; i++)
            {
                for (int j = 0; j < y; j++)
                {
                    naujasmasyvas[i, j] = originalas[i, j];
                }
            }
            originalas = naujasmasyvas;
        }
        public static T masyvo_antro_stulpelio_max_2d<T>(ref T[,] originalas,int indeksas)
        {
            T max;
            int x = originalas.GetLength(1);
            var naujas_masyvas = new T[x];
            for (int i = 0; i < x; i++)
            {
                naujas_masyvas[i] = originalas[indeksas, i];
            }
            max=naujas_masyvas.Max();
            return max;
        }
        public static void masyvo_dydzio_keitimas_3d<T>(ref T[,,,] originalas, int nx, int ny, int nz)
        {
            var naujasmasyvas = new T[nx, ny, nz];
            int x = Math.Min(nx, originalas.GetLength(0));
            int y = Math.Min(ny, originalas.GetLength(1));
            int z = Math.Min(nz, originalas.GetLength(2));
            for (int i = 0; i < x; i++)
            {
                for (int j = 0; j < y; j++)
                {
                    for(int k=0;k< z; k++)
                    {
                        naujasmasyvas[i, j, k] = originalas[i, j, k];
                    }
                }
            }
            originalas = naujasmasyvas;
        }
        public static void tuscia_texbox_skaicius(string tekstas,string patikrintas_tekstas, bool salyga, bool
stabdis)
        {
            if (string.IsNullOrEmpty(tekstas))
            {
                tekstas = patikrintas_tekstas;
                salyga = true;
                stabdis = false;
            }
        }
    }
}
```



```

Globalus_kintamieji.posistemiu_iejimu_pavadinimu_masyvas = new string[1, 1];
Globalus_kintamieji.posistemiu_iejimu_diapozonu_min_masyvas = new double[1, 1];
Globalus_kintamieji.posistemiu_iejimu_diapozonu_max_masyvas = new double[1, 1];
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_sk_masyvas = new int[1, 1];
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_pavadinimu_masyvas = new string[1,1,1];
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas = new double[1, 1, 1];
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas = new double[1, 1, 1];
Globalus_kintamieji.posistemiu_isejimu_pavadinimu_masyvas = new string[1];
Globalus_kintamieji.posistemiu_isejimu_diapozonu_min_masyvas = new double[1];
Globalus_kintamieji.posistemiu_isejimu_diapozonu_max_masyvas = new double[1];
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_sk_masyvas = new int[1];
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_pavadinimu_masyvas = new string[1, 1];
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas = new double[1, 1];
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraiskumu_masyvas = new double[1, 1];
Globalus_kintamieji.posistemiu_isoriniu_duomenu_tekstinis_masyvas = new string[1];
Globalus_kintamieji.posistemiu_isoriniu_duomenu_masyvas = new double[1, 1, 1];
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas = new int[1];
Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas = new string[1, 1];
Globalus_kintamieji.posistemiu_taisykliu_aktyviu_iejimu_indeksu_masyvas = new string[1, 1, 1];
Globalus_kintamieji.posistemiu_taisykliu_vidurkiu_masyvas = new double[1, 1, 1];
Globalus_kintamieji.posistemiu_taisykliu_neraiskumu_masyvas = new double[1, 1, 1];
Globalus_kintamieji.posistemiu_isejimu_taisykliu_vidurkiu_masyvas = new double[1, 1];
Globalus_kintamieji.posistemiu_isejimu_taisykliu_neraiskumu_masyvas = new double[1, 1];
Globalus_kintamieji.posistemiu_rezultatu_masyvas = new double[1,1];
Globalus_kintamieji.posistemiu_rezultatu_veleveliu_masyvas = new string[1];

foreach (Control control in this.Controls)
{
    control.PreviewKeyDown += new PreviewKeyDownEventHandler(control_PreviewKeyDown);
}
}
void control_PreviewKeyDown(object sender, PreviewKeyDownEventArgs e)
{
    if (e.KeyCode == Keys.Up || e.KeyCode == Keys.Down || e.KeyCode == Keys.Left || e.KeyCode ==
Keys.Right)
    {
        e.IsInputKey = true;
    }
}
private void prideti_posisteme_ToolStripMenuItem_Click(object sender, EventArgs e)
{
    mygtuku_masyvo_dydis++;
    int indeksas = mygtuku_masyvo_dydis - 1;
    var koordinates = this.PointToClient(Cursor.Position);
    Array.Resize(ref mygtuku_masyvas, mygtuku_masyvo_dydis);
    Array.Resize(ref Globalus_kintamieji.posistemiu_pavadinimu_masyvas, mygtuku_masyvo_dydis);
    Array.Resize(ref
Globalus_kintamieji.posistemiu_isoriniu_duomenu_tekstinis_masyvas,mygtuku_masyvo_dydis);
    Array.Resize(ref Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas, mygtuku_masyvo_dydis);
    Funkcijos.masyvo_dydzio_keitimas_3d(ref Globalus_kintamieji.posistemiu_isoriniu_duomenu_masyvas,
mygtuku_masyvo_dydis, 1, 1);
    int skaicius = mygtuku_masyvo_dydis + istrintu_mygtuku_skaicius;
    Globalus_kintamieji.posistemiu_pavadinimu_masyvas[indeksas] = "Posistemė " + skaicius;
    Array.Resize(ref Globalus_kintamieji.posistemiu_x_koordinaciu_masyvas, mygtuku_masyvo_dydis);
    Array.Resize(ref Globalus_kintamieji.posistemiu_y_koordinaciu_masyvas, mygtuku_masyvo_dydis);
    Array.Resize(ref Globalus_kintamieji.posistemiu_isejimu_pavadinimu_masyvas, mygtuku_masyvo_dydis);
    Array.Resize(ref Globalus_kintamieji.posistemiu_isejimu_diapozonu_min_masyvas,
mygtuku_masyvo_dydis);
    Array.Resize(ref Globalus_kintamieji.posistemiu_isejimu_diapozonu_max_masyvas,
mygtuku_masyvo_dydis);
    Array.Resize(ref Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_sk_masyvas,
mygtuku_masyvo_dydis);
    if (Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas.Max() > 0)
    {
        Funkcijos.masyvo_dydzio_keitimas_2d(ref
Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas, mygtuku_masyvo_dydis,
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas.Max());
    }
    else
    {
        Funkcijos.masyvo_dydzio_keitimas_2d(ref
Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas, mygtuku_masyvo_dydis, 1);
    }
    int temp_koordinates_X = koordinates.X / atstumas_tarp_posistemiu_X;
    int temp_koordinates_Y = 0;
    for (int i = 0; i < mygtuku_masyvo_dydis; i++)

```

```

    {
        for (int j = 0; j < mygtuku_masyvo_dydis; j++)
        {
            if (mygtuku_masyvo_dydis != 1)
            {
                if (temp_koordinates_X == Globalus_kintamieji.posistemiu_x_koordinaciju_masyvas[j] &&
temp_koordinates_Y == Globalus_kintamieji.posistemiu_y_koordinaciju_masyvas[j])
                {
                    temp_koordinates_Y++;
                }
            }
        }
        Globalus_kintamieji.posistemiu_x_koordinaciju_masyvas[indeksas] = temp_koordinates_X;
        Globalus_kintamieji.posistemiu_y_koordinaciju_masyvas[indeksas] = temp_koordinates_Y;
        var m = new Button();
        m.Size = new System.Drawing.Size(posistemės_dydis_X, posistemės_dydis_Y);
        m.UseVisualStyleBackColor = true;
        m.Location = new Point(Globalus_kintamieji.posistemiu_x_koordinaciju_masyvas[indeksas] *
atstumas_tarp_posistemiu_X + objekto_atstumas_is_kaires,
Globalus_kintamieji.posistemiu_y_koordinaciju_masyvas[indeksas] * atstumas_tarp_posistemiu_Y +
objekto_atstumas_i_apacia);
        m.Text = Globalus_kintamieji.posistemiu_pavadinimu_masyvas[indeksas];
        m.TextAlign = ContentAlignment.TopLeft;
        m.Tag = indeksas;
        m.Click += new EventHandler(posistemės_paspaudimas);
        m.Enter += new EventHandler(mygtuko_aktyvavimo_ivykis);
        m.Leave += new EventHandler(mygtuko_deaktyvavimo_ivykis);
        m.MouseDown += new MouseEventHandler(mygtuko_paspaudimo_ivykis);
        m.MouseMove += new MouseEventHandler(mygtuko_slinkimo_ivykis);
        m.MouseUp += new MouseEventHandler(mygtuko_padejimo_ivykis);
        m.PreviewKeyDown += new PreviewKeyDownEventHandler(control_PreviewKeyDown);
        m.KeyDown += new KeyEventHandler(maygtuko_istrinimo_ivykis);
        m.ContextMenuStrip = mygtuko_contextMenuStrip1;
        mygtuku_masyvas[indeksas] = m;
        this.Controls.Add(m);
    }
    bool leidimas_istrinti_mygtukas=true;
    private void maygtuko_istrinimo_ivykis(object sender, KeyEventArgs e)
    {
        var m = sender as Button;
        if (e.KeyCode == Keys.Delete&&!leidimas)
        {
            leidimas = false;
            leidimas_istrinti_mygtukas = true;
            foreach (var linija in linijos)
            {
                if ((m.Location.X + 100) == linija.Start.X && (m.Location.Y + 35) == linija.Start.Y ||
m.Location.X == linija.End.X && (m.Location.Y + 35) == linija.End.Y)
                {
                    leidimas_istrinti_mygtukas = false;
                }
            }
            if (leidimas_istrinti_mygtukas)
            {
                this.Controls.Remove(m);
                Funkcijos.masyvo_elemento_istrinimas(ref mygtuku_masyvas,int.Parse(m.Tag.ToString()));
                Funkcijos.masyvo_elemento_istrinimas(ref Globalus_kintamieji.posistemiu_pavadinimu_masyvas,
int.Parse(m.Tag.ToString()));
                Funkcijos.masyvo_elemento_istrinimas(ref
Globalus_kintamieji.posistemiu_x_koordinaciju_masyvas, int.Parse(m.Tag.ToString()));
                Funkcijos.masyvo_elemento_istrinimas(ref
Globalus_kintamieji.posistemiu_y_koordinaciju_masyvas, int.Parse(m.Tag.ToString()));

                for(int i = 0; i < mygtuku_masyvas.Length; i++)
                {
                    mygtuku_masyvas[i].Tag = i;
                }
                mygtuku_masyvo_dydis--;
                istrintu_mygtuku_skaicius++;
                leidimas = true;
            }
        }
    }
    if(e.KeyCode==Keys.Z&& ModifierKeys.HasFlag(Keys.Control))
    {
        prideti_posisteme_ToolStripMenuItem_Click(sender, e);
    }
}

```

```

    }
}
int paspaudimas = 0;
int liniju_indeksas = 0;
Point pt1;
Point pt2;
int pt1_X, pt1_Y, pt2_X, pt2_Y;
private void posistemės_paspaudimas(object sender, EventArgs e)
{
    var m = sender as Button;
    if (leidimas_redaguoti_posisteme)
    {
        Globalus_kintamieji.mygtuko_tagas = int.Parse(m.Tag.ToString());
        Posistemės_dialogas fd = new Posistemės_dialogas();
        fd.Location = Cursor.Position;
        fd.ShowDialog();
        leidimas_perkelti = true;
        leidimas_redaguoti_posisteme = false;
    }
    if (leidimas_pervadinti_mygtuka)
    {
        Globalus_kintamieji.mygtuko_velele = true;
        Globalus_kintamieji.mygtuko_tagas = int.Parse(m.Tag.ToString());
        Pervadinti f = new Pervadinti();
        f.Location = Cursor.Position;
        f.ShowDialog();
        mygtuku_masyvas[Globalus_kintamieji.mygtuko_tagas].Text =
Globalus_kintamieji.posistemiu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas];
        Globalus_kintamieji.mygtuko_velele = false;
        leidimas_perkelti = true;
        leidimas_pervadinti_mygtuka = false;
    }
    if (sujungti == true)
    {
        paspaudimas++;
        if (paspaudimas % 2 != 0)
        {
            if (!tarpinis_stabdis_SS)
            {
                pt1_X = m.Location.X / atstumas_tarp_posistemiu_X;
                pt1_Y = m.Location.Y / atstumas_tarp_posistemiu_Y;
            }
            else
            {
                if (!tarpinis_stabdis_SS)
                {
                    pt2_X = m.Location.X / atstumas_tarp_posistemiu_X;
                    pt2_Y = m.Location.Y / atstumas_tarp_posistemiu_Y;
                }
                if (pt1_X > pt2_X || (pt2_X - pt1_X) != 1)
                {
                    MessageBox.Show("Posistemės sujungiamos tik iš žemesnio į aukštesnį lygį");
                    pt1_X = -1;
                    pt1_Y = -1;
                    pt2_X = -1;
                    pt2_Y = -1;
                    sujungti = false;
                    leidimas_perkelti = true;
                }
                else if (pt1_X==pt2_X)
                {
                    MessageBox.Show("Neįsivaizduoju, ką tu norite pasiekti");
                    pt1_X = -1;
                    pt1_Y = -1;
                    pt2_X = -1;
                    pt2_Y = -1;
                    sujungti = false;
                    leidimas_perkelti = true;
                }
            }
            else
            {
                pt1.X = pt1_X * atstumas_tarp_posistemiu_X + objekto_atstumas_is_kaires + 100;
                pt1.Y = pt1_Y * atstumas_tarp_posistemiu_Y + objekto_atstumas_i_apacia + 35;
                pt2.X = pt2_X * atstumas_tarp_posistemiu_X + objekto_atstumas_is_kaires;
                pt2.Y = pt2_Y * atstumas_tarp_posistemiu_Y + objekto_atstumas_i_apacia + 35;
            }
        }
    }
}

```

```

        foreach (var tiese in linijos)
        {
            if (tiese.Start.X == pt1.X && tiese.Start.Y == pt1.Y && tiese.End.X == pt2.X &&
                tiese.End.Y == pt2.Y)
            {
                {
                    pt1_X = -1;
                    pt1_Y = -1;
                    pt2_X = -1;
                    pt2_Y = -1;
                    sujungti = false;
                    leidimas_perkelti = true;
                    MessageBox.Show("Sujungimo linijos negali persikloti");
                }
            }
            var linija = new Line();
            linija.Color = Color.Black;
            linijos.Add(linija);
            linija.Start = pt1;
            linija.End = pt2;
            var kelias = new GraphicsPath();
            kelias.AddLine(linija.Start, linija.End);
            keliai.Add(kelias);
            this.Invalidate();
            sujungti = false;
            leidimas_perkelti = true;
            liniju_indeksas++;
        }
    }
}
if (leidimas_skaiciuoti_pasirinkta)
{
    double[, ] laikinas_duomenu_masyvas = new double[1, 1, 1];
    bool leidimas_skaiciuoti = false;
    bool rasta_klaidu = false;
    int liniju_skaicius = 0;
    foreach(var linija in linijos)
    {
        if(m.Location.X == linija.End.X && (m.Location.Y + 35) == linija.End.Y)
        {
            foreach(var mygtukas in mygtuku_masyvas)
            {
                if((mygtukas.Location.X + 100) == linija.Start.X && (mygtukas.Location.Y + 35) ==
                    linija.Start.Y)
                {
                    if
                        (Globalus_kintamieji.posistemiu_rezultatu_veleveliu_masyvas[int.Parse(mygtukas.Tag.ToString())] == "done")
                    {
                        Funkcijos.masyvo_dydzio_keitimas_3d(ref laikinas_duomenu_masyvas,
                            Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length,
                            Globalus_kintamieji.posistemiu_rezultatu_masyvas.GetLength(1), liniju_skaicius+1);
                        for(int i=0;i<
                            Globalus_kintamieji.posistemiu_rezultatu_masyvas.GetLength(1); i++)
                        {
                            laikinas_duomenu_masyvas[int.Parse(m.Tag.ToString()), i,
                                liniju_skaicius] = Globalus_kintamieji.posistemiu_rezultatu_masyvas[int.Parse(mygtukas.Tag.ToString()), i];
                        }
                        leidimas_skaiciuoti = true;
                        liniju_skaicius++;
                    }
                    else
                    {
                        MessageBox.Show("Neapskaičiuota Posistemė: " +
                            Globalus_kintamieji.posistemiu_pavadinimu_masyvas[int.Parse(mygtukas.Tag.ToString())]);
                        rasta_klaidu = true;
                    }
                }
            }
        }
    }
}
if
(Globalus_kintamieji.posistemiu_isoriniu_duomenu_tekstinis_masyvas[int.Parse(m.Tag.ToString())] != null)
{
    Funkcijos.masyvo_dydzio_keitimas_3d(ref laikinas_duomenu_masyvas,
        Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length,
        Globalus_kintamieji.posistemiu_isoriniu_duomenu_masyvas.GetLength(1), liniju_skaicius + 1 +
        Globalus_kintamieji.posistemiu_isoriniu_duomenu_masyvas.GetLength(2));
}
}

```



```

        for (int i = 0; i < Globalus_kintamieji.posistemiu_isoriniu_duomenu_masyvas.GetLength(1);
i++)
    {
        for (int j = 0; j <
Globalus_kintamieji.posistemiu_isoriniu_duomenu_masyvas.GetLength(2); j++)
        {
            laikinas_duomenu_masyvas[int.Parse(m.Tag.ToString()), i, liniju_skaicius+j] =
Globalus_kintamieji.posistemiu_isoriniu_duomenu_masyvas[int.Parse(m.Tag.ToString()), i, j];
        }
    }
    if (liniju_skaicius == 0)
    {
        leidimas_skaiciuoti = true;
    }
    if (leidimas_skaiciuoti&&!rasta_klaidu)
    {
        Array.Resize(ref Globalus_kintamieji.posistemiu_rezultatu_veleveliu_masyvas,
Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length);
        Funkcijos.masyvo_dydzio_keitimas_2d(ref Globalus_kintamieji.posistemiu_rezultatu_masyvas,
Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length,
Globalus_kintamieji.posistemiu_isoriniu_duomenu_masyvas.GetLength(1));
        for(int i=0;i< Globalus_kintamieji.posistemiu_isoriniu_duomenu_masyvas.GetLength(1); i++)
        {
            double galutinis = 0;
            double galutinis_be = 0;
            for(int j = 0; j <
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas[int.Parse(m.Tag.ToString());] j++)
            {
                double mikro = 1;
                for(int k = 0; k <
Globalus_kintamieji.posistemiu_iejimu_skaicius[int.Parse(m.Tag.ToString());] k++)
                {
                    if
(Globalus_kintamieji.posistemiu_taisykliu_aktyviu_iejimu_indeksu_masyvas[int.Parse(m.Tag.ToString()), j, k] ==
"on")
                    {
                        mikro = mikro * Math.Exp((-
(laikinas_duomenu_masyvas[int.Parse(m.Tag.ToString()), i, k] -
Globalus_kintamieji.posistemiu_taisykliu_vidurkiu_masyvas[int.Parse(m.Tag.ToString()), j, k]) *
(laikinas_duomenu_masyvas[int.Parse(m.Tag.ToString()), i, k] -
Globalus_kintamieji.posistemiu_taisykliu_vidurkiu_masyvas[int.Parse(m.Tag.ToString()), j, k])) / (2 *
(Globalus_kintamieji.posistemiu_taisykliu_neraiskumu_masyvas[int.Parse(m.Tag.ToString()), j, k] *
Globalus_kintamieji.posistemiu_taisykliu_neraiskumu_masyvas[int.Parse(m.Tag.ToString()), j, k]) +
Globalus_kintamieji.e));
                    }
                }
                galutinis_be = galutinis_be + ((mikro) /
(Globalus_kintamieji.posistemiu_isejimu_taisykliu_neraiskumu_masyvas[int.Parse(m.Tag.ToString()), j] *
Globalus_kintamieji.posistemiu_isejimu_taisykliu_neraiskumu_masyvas[int.Parse(m.Tag.ToString()), j] +
Globalus_kintamieji.e));
                galutinis = galutinis +
((Globalus_kintamieji.posistemiu_isejimu_taisykliu_vidurkiu_masyvas[int.Parse(m.Tag.ToString()), j] * mikro) /
(Globalus_kintamieji.posistemiu_isejimu_taisykliu_neraiskumu_masyvas[int.Parse(m.Tag.ToString()), j] *
Globalus_kintamieji.posistemiu_isejimu_taisykliu_neraiskumu_masyvas[int.Parse(m.Tag.ToString()), j] +
Globalus_kintamieji.e));
            }
            Globalus_kintamieji.posistemiu_rezultatu_masyvas[int.Parse(m.Tag.ToString()), i] =
galutinis / galutinis_be;
        }
        Globalus_kintamieji.posistemiu_rezultatu_veleveliu_masyvas[int.Parse(m.Tag.ToString())] =
"done";
    }
    if (solo)
    {
        Globalus_kintamieji.mygtuko_tagas = int.Parse(m.Tag.ToString());
        Rezultato_dialogas Rd = new Rezultato_dialogas();
        Rd.Location = Cursor.Position;
        Rd.ShowDialog();
        solo = false;
    }
}
leidimas_skaiciuoti_pasirinkta = false;
leidimas_perkelti = true;
}
if (leidimas_rezultatui)
{
    Globalus_kintamieji.mygtuko_tagas = int.Parse(m.Tag.ToString());
}

```

```

        Rezultato_dialogas Rd = new Rezultato_dialogas();
        Rd.Location = Cursor.Position;
        Rd.ShowDialog();
        leidimas_rezultatui = false;
        leidimas_perkelti = true;
    }
}
private void mygtuko_aktyvavimo_ivykis(object sender, EventArgs e)
{
    var m = sender as Button;
    m.BackColor = Color.MediumSeaGreen;
    leidimas_istrinti_sujungima = false;
}
private void mygtuko_deaktyvavimo_ivykis(object sender, EventArgs e)
{
    var m = sender as Button;
    m.BackColor = Color.Gainsboro;
}
bool leidimas = false;
private void mygtuko_paspaudimo_ivykis(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left)
    {
        var m = sender as Button;
        _Offset = new Point(e.X, e.Y);
        pradine_x = m.Location.X;
        pradine_y = m.Location.Y;
        m.BringToFront();
        leidimas = true;
        foreach (var linija in linijos)
        {
            if ((m.Location.X+100) == linija.Start.X && (m.Location.Y+35) == linija.Start.Y ||
m.Location.X == linija.End.X && (m.Location.Y +35) == linija.End.Y)
            {
                leidimas = false;
            }
        }
    }
}
private void mygtuko_slinkimo_ivykis(object sender, MouseEventArgs e)
{
    if (leidimas == true && leidimas_perkelti == true)
    {
        if (_Offset != Point.Empty)
        {
            var m = sender as Button;
            Point newlocation = m.Location;
            newlocation.X += e.X - _Offset.X;
            newlocation.Y += e.Y - _Offset.Y;
            m.Location = newlocation;
        }
    }
}
bool leidimas_pozicijoms_apsikeisti;
private void mygtuko_padejimo_ivykis(object sender, MouseEventArgs e)
{
    if (leidimas == true && leidimas_perkelti == true)
    {
        var m = sender as Button;
        leidimas_pozicijoms_apsikeisti = true;
        foreach (var linija in linijos)
        {
            if ((m.Location.X+100) / atstumas_tarp_posistemiu_X == linija.Start.X /
atstumas_tarp_posistemiu_X && m.Location.Y / atstumas_tarp_posistemiu_Y == (linija.Start.Y-
35)/atstumas_tarp_posistemiu_Y || m.Location.X / atstumas_tarp_posistemiu_X == linija.End.X /
atstumas_tarp_posistemiu_X && m.Location.Y / atstumas_tarp_posistemiu_Y == (linija.End.Y-35) /
atstumas_tarp_posistemiu_Y)
            {
                leidimas_pozicijoms_apsikeisti = false;
            }
        }
        if (leidimas_pozicijoms_apsikeisti == true)
        {
            m.Location = new Point((m.Location.X / atstumas_tarp_posistemiu_X) *
atstumas_tarp_posistemiu_X + objekto_atstumas_is_kaires, (m.Location.Y / atstumas_tarp_posistemiu_Y) *
atstumas_tarp_posistemiu_Y + objekto_atstumas_i_apacia);
            for (int i = 0; i < mygtuku_masyvas.GetLength(0); i++)

```

```

        {
            if (Globalus_kintamieji.posistemiu_x_koordinaciju_masyvas[i] == m.Location.X /
atstumas_tarp_posistemiu_X && Globalus_kintamieji.posistemiu_y_koordinaciju_masyvas[i] == m.Location.Y /
atstumas_tarp_posistemiu_Y)
            {
                Globalus_kintamieji.posistemiu_x_koordinaciju_masyvas[i] = pradine_x /
atstumas_tarp_posistemiu_X;
                Globalus_kintamieji.posistemiu_y_koordinaciju_masyvas[i] = pradine_y /
atstumas_tarp_posistemiu_Y;
                mygtuku_masyvas[i].Location = new
Point(Globalus_kintamieji.posistemiu_x_koordinaciju_masyvas[i] * atstumas_tarp_posistemiu_X +
objekto_atstumas_is_kaires, Globalus_kintamieji.posistemiu_y_koordinaciju_masyvas[i] * atstumas_tarp_posistemiu_Y
+ objekto_atstumas_i_apacia);
            }
            Globalus_kintamieji.posistemiu_x_koordinaciju_masyvas[Int32.Parse(m.Tag.ToString())] =
m.Location.X / atstumas_tarp_posistemiu_X;
            Globalus_kintamieji.posistemiu_y_koordinaciju_masyvas[Int32.Parse(m.Tag.ToString())] =
m.Location.Y / atstumas_tarp_posistemiu_Y;
        }
        else
        {
            m.Location = new Point((pradine_x / atstumas_tarp_posistemiu_X) * atstumas_tarp_posistemiu_X
+ objekto_atstumas_is_kaires, (pradine_y / atstumas_tarp_posistemiu_Y) * atstumas_tarp_posistemiu_Y +
objekto_atstumas_i_apacia);
            leidimas = false;
            leidimas_pozicijoms_apsikeisti = true;
            _Offset = Point.Empty;
        }
    }
}
private void prideti_lygio_pavadinima_ToolStripMenuItem_Click(object sender, EventArgs e)
{
    antrasciu_masyvo_dydis++;
    int indeksas = antrasciu_masyvo_dydis - 1;
    Array.Resize(ref antrasciu_masyvas, antrasciu_masyvo_dydis);
    Array.Resize(ref Globalus_kintamieji.antrasciu_pavadinimu_masyvas, antrasciu_masyvo_dydis);
    Array.Resize(ref Globalus_kintamieji.antrasciu_x_koordinaciju_masyvas, antrasciu_masyvo_dydis);
    var koordinates = this.PointToClient(Cursor.Position);
    int temp_koordinates_X = koordinates.X / atstumas_tarp_posistemiu_X;
    int temp_koordinates_Y = 30;
    for (int i = 0; i < antrasciu_masyvo_dydis; i++)
    {
        for (int j = 0; j < antrasciu_masyvo_dydis; j++)
        {
            if (antrasciu_masyvo_dydis != 1)
            {
                if (temp_koordinates_X == Globalus_kintamieji.antrasciu_x_koordinaciju_masyvas[j])
                {
                    temp_koordinates_X++;
                }
            }
        }
    }
    Globalus_kintamieji.antrasciu_x_koordinaciju_masyvas[indeksas] = temp_koordinates_X;
    Globalus_kintamieji.antrasciu_pavadinimu_masyvas[indeksas] = "Lygis " + antrasciu_masyvo_dydis;
    var l = new ReadOnlyTextBox();
    //l.ReadOnly = true;
    l.BorderStyle = BorderStyle.None;
    l.BackColor = Color.Gainsboro;
    l.Location = new Point(Globalus_kintamieji.antrasciu_x_koordinaciju_masyvas[indeksas] *
atstumas_tarp_posistemiu_X + objekto_atstumas_is_kaires, temp_koordinates_Y);
    l.Text = Globalus_kintamieji.antrasciu_pavadinimu_masyvas[indeksas];
    l.Tag = indeksas;
    l.DoubleClick += new EventHandler(antrastes_pervadinimo_ivykis);
    l.MouseDown += new MouseEventHandler(antrastes_paspaudimo_ivykis);
    l.MouseMove += new MouseEventHandler(antrastes_slinkimo_ivykis);
    l.MouseUp += new MouseEventHandler(antrastes_padejimo_ivykis);
    l.PreviewKeyDown += new PreviewKeyDownEventHandler(control_PreviewKeyDown);
    l.KeyDown += new KeyEventHandler(antrastes_klaviso_ivykis);
    antrasciu_masyvas[indeksas] = 1;
    this.Controls.Add(l);
}
private void antrastes_klaviso_ivykis(object sender, KeyEventArgs e)
{
    var l = sender as ReadOnlyTextBox;
    if (e.KeyCode == Keys.Delete&&!Dragging)

```

```

    {
        this.Controls.Remove(l);
        Funkcijos.masyvo_elemento_istrinimas(ref antrasciu_masyvas, int.Parse(l.Tag.ToString()));
        Funkcijos.masyvo_elemento_istrinimas(ref Globalus_kintamieji.antrasciu_pavadinimu_masyvas,
int.Parse(l.Tag.ToString()));
        Funkcijos.masyvo_elemento_istrinimas(ref Globalus_kintamieji.antrasciu_x_koordinaciju_masyvas,
int.Parse(l.Tag.ToString()));
        antrasciu_masyvo_dydis--;
        for (int i = 0; i < antrasciu_masyvas.Length; i++)
        {
            antrasciu_masyvas[i].Tag = i;
        }
    }
    if(e.KeyCode==Keys.A&& ModifierKeys.HasFlag(Keys.Control))
    {
        prideti_lygio_pavadinima_ToolStripMenuItem_Click(sender, e);
    }
}
bool Dragging = false;
Point DragStart = Point.Empty;
private void antrastes_paspaudimo_ivykis(object sender, MouseEventArgs e)
{
    var l = sender as TextBox;
    Dragging = true;
    DragStart = new Point(e.X, e.Y);
    l.Capture = true;
    pradine_x = l.Location.X;
    pradine_y = l.Location.Y;
    l.BringToFront();
    leidimas_istrinti_sujungima = false;
}
private void antrastes_slinkimo_ivykis(object sender, MouseEventArgs e)
{
    var l = sender as TextBox;
    if (Dragging)
    {
        l.Left = Math.Max(0, e.X + l.Left - DragStart.X);
        l.Top = Math.Max(0, e.Y + l.Top - DragStart.Y);
    }
}
private void antrastes_padejimo_ivykis(object sender, MouseEventArgs e)
{
    var l = sender as TextBox;
    Dragging = false;
    l.Capture = false;
    l.Location = new Point((l.Location.X / atstumas_tarp_posistemiu_X) * atstumas_tarp_posistemiu_X +
objekto_atstumas_is_kaires, 30);
    for (int i = 0; i < antrasciu_masyvo_dydis; i++)
    {
        if (Globalus_kintamieji.antrasciu_x_koordinaciju_masyvas[i] == l.Location.X /
atstumas_tarp_posistemiu_X)
        {
            Globalus_kintamieji.antrasciu_x_koordinaciju_masyvas[i] = pradine_x /
atstumas_tarp_posistemiu_X;
            antrasciu_masyvas[i].Location = new
Point(Globalus_kintamieji.antrasciu_x_koordinaciju_masyvas[i] * atstumas_tarp_posistemiu_X +
objekto_atstumas_is_kaires, 30);
        }
        Globalus_kintamieji.antrasciu_x_koordinaciju_masyvas[Int32.Parse(l.Tag.ToString())] = l.Location.X /
atstumas_tarp_posistemiu_X;
    }
}
private void antrastes_pervadinimo_ivykis(object sender, EventArgs e)
{
    var l=sender as TextBox;
    Globalus_kintamieji.antrastes_veleleve = true;
    Globalus_kintamieji.antrastes_tagas = int.Parse(l.Tag.ToString());
    Pervadinti f= new Pervadinti();
    f.Location = Cursor.Position;
    f.ShowDialog();
    antrasciu_masyvas[Globalus_kintamieji.antrastes_tagas].Text=
Globalus_kintamieji.antrasciu_pavadinimu_masyvas[Globalus_kintamieji.antrastes_tagas];
    Globalus_kintamieji.antrastes_veleleve = false;
}
bool sujungti = false;
Point linijos_pradzia = Point.Empty;
bool leidimas_perkelti = true;

```

```

private void sujungtiNuoToolStripMenuItem_Click(object sender, EventArgs e)
{
    leidimas_redaguoti_posisteme = false;
    leidimas_pervadinti_mygtuka = false;
    leidimas_skaiciuoti_pasirinkta = false;
    leidimas_rezultatui = false;
    sujungti = true;
    leidimas_perkelti = false;
}
private void FuzzyThis_Paint(object sender, PaintEventArgs e)
{
    // Reset the background.
    e.Graphics.Clear(Color.Gainsboro);
    e.Graphics.SmoothingMode = SmoothingMode.AntiAlias;
    foreach (var line in linijos)
    {
        using (var pen = new Pen(line.Color, line.Width))
        {
            e.Graphics.DrawLine(pen, line.Start, line.End);
        }
    }
}
PointF a = new PointF();
PointF b = new PointF();
bool leidimas_istrinti_sujungima = false;
private void FuzzyThis_MouseDown(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left)
    {
        if (keliai != null)
        {
            foreach (var kelias in keliai)
            {
                if (kelias.IsOutlineVisible(e.X, e.Y, new Pen(Color.Black, 8 /*tolerance*/))
                {
                    leidimas_istrinti_sujungima = true;
                    a = kelias.PathData.Points.First();
                    b = kelias.PathData.Points.Last();
                    foreach (var line in linijos)
                    {
                        if (line.Start == a && line.End == b)
                        {
                            line.Width = 3;
                            line.Color = Color.MediumSeaGreen;
                        }
                        else
                        {
                            line.Width = 1;
                            line.Color = Color.Black;
                        }
                    }
                    this.Invalidate();
                }
            }
        }
    }
}
if (e.Button == MouseButtons.Left && ModifierKeys.HasFlag(Keys.Control))
{
    if (leidimas_istrinti_sujungima)
    {
        int linija_i = 0, kelias_j = 0;
        int linijos_indeksas = 0, kelio_indeksas = 0;
        foreach (var linija in linijos)
        {
            if (linija.Width == 3 && linija.Color == Color.MediumSeaGreen)
            {
                linijos_indeksas = linija_i;
                foreach (var kelias in keliai)
                {
                    a = kelias.PathData.Points.First();
                    b = kelias.PathData.Points.Last();
                    if (linija.Start == a && linija.End == b)
                    {
                        kelio_indeksas = kelias_j;
                    }
                    kelias_j++;
                }
            }
        }
    }
}

```

```

        }
        linija_i++;
    }
    linijos.RemoveAt(linijos_indeksas);
    keliai.RemoveAt(kelio_indeksas);
    this.Invalidate();
    leidimas_istrinti_sujungima = false;
}
}
}
private void uzdaryti_ToolStripMenuItem_Click(object sender, EventArgs e)
{
    Application.Exit();
}
bool leidimas_pervadinti_mygtuka=false;
private void pervadintiToolStripMenuItem_Click(object sender, EventArgs e)
{
    leidimas_redaguoti_posisteme = false;
    sujungti = false;
    leidimas_skaiciuoti_pasirinkta = false;
    leidimas_rezultatui = false;
    leidimas_pervadinti_mygtuka = true;
    leidimas_perkelti = false;
}
bool leidimas_redaguoti_posisteme=false;
private void redaguotiToolStripMenuItem_Click(object sender, EventArgs e)
{
    leidimas_pervadinti_mygtuka = false;
    sujungti = false;
    leidimas_skaiciuoti_pasirinkta = false;
    leidimas_rezultatui = false;
    leidimas_redaguoti_posisteme = true;
    leidimas_perkelti = false;
}
bool leidimas_skaiciuoti_pasirinkta;
bool solo = false;
private void skaiciuoti_pasirinkta_ToolStripMenuItem_Click(object sender, EventArgs e)
{
    leidimas_pervadinti_mygtuka = false;
    sujungti = false;
    leidimas_redaguoti_posisteme = false;
    leidimas_perkelti = false;
    leidimas_rezultatui = false;
    leidimas_skaiciuoti_pasirinkta = true;
    solo = true;
}
bool leidimas_rezultatui;
private void rezultatasToolStripMenuItem_Click(object sender, EventArgs e)
{
    leidimas_pervadinti_mygtuka = false;
    sujungti = false;
    leidimas_redaguoti_posisteme = false;
    leidimas_perkelti = false;
    leidimas_skaiciuoti_pasirinkta = false;
    leidimas_rezultatui = true;
}
private void skaiciuoti_visas_ToolStripMenuItem_Click(object sender, EventArgs e)
{
    for(int i=0;i< Globalus_kintamieji.posistemiu_x_koordinaciu_masyvas.Max(); i++)
    {
        for(int j = 0; j < Globalus_kintamieji.posistemiu_y_koordinaciu_masyvas.Max(); j++)
        {
            for(int k = 0; k < mygtuku_masyvas.Length; k++)
            {
                if (Globalus_kintamieji.posistemiu_x_koordinaciu_masyvas[k] == i &&
Globalus_kintamieji.posistemiu_y_koordinaciu_masyvas[k] == j)
                {
                    leidimas_skaiciuoti_pasirinkta = true;
                    if (i == Globalus_kintamieji.posistemiu_x_koordinaciu_masyvas.Max() && j ==
Globalus_kintamieji.posistemiu_y_koordinaciu_masyvas.Max())
                    {
                        solo = true;
                    }
                    mygtuku_masyvas[k].PerformClick();
                }
            }
        }
    }
}

```

```

    }
}

private void paprasta_sistema_toolStripMenuItem_Click(object sender, EventArgs e)
{
    if (mygtuku_masyvo_dydis==0)
    {
        Point a = Cursor.Position;
        Cursor.Position = this.PointToScreen(new Point(0, 0));
        prideti_lygio_pavadinima_ToolStripMenuItem_Click(sender, e);
        prideti_posisteme_ToolStripMenuItem_Click(sender, e);
        Cursor.Position = a;
    }
}

bool tarpinis_stabdis_SS=false;
private void tarpines_linijos_SS(object sender, EventArgs e,int a, int b, int c, int d, int indeksas)
{
    sujungti = true;
    tarpinis_stabdis_SS = true;
    pt1_X = a;
    pt1_Y = b;
    pt2_X = c;
    pt2_Y = d;
    posistemės_paspaudimas(sender, e);
    posistemės_paspaudimas(sender, e);
    tarpinis_stabdis_SS = false;
    sujungti = false;
}

private void sudetinga_sistema_toolStripMenuItem_Click(object sender, EventArgs e)
{
    if (mygtuku_masyvo_dydis==0)
    {
        Point a = Cursor.Position;
        Cursor.Position = this.PointToScreen(new Point(0, 0));
        prideti_lygio_pavadinima_ToolStripMenuItem_Click(sender, e);
        Cursor.Position = this.PointToScreen(new Point(250, 0));
        prideti_lygio_pavadinima_ToolStripMenuItem_Click(sender, e);
        Cursor.Position = this.PointToScreen(new Point(450, 0));
        prideti_lygio_pavadinima_ToolStripMenuItem_Click(sender, e);
        for (int i = 0; i < 7; i++)
        {
            if (i < 4)
            {
                Cursor.Position = this.PointToScreen(new Point(0, 0));
                prideti_posisteme_ToolStripMenuItem_Click(sender, e);
                if (i < 2)
                {
                    tarpines_linijos_SS(sender, e, 0, i, 1, 0, liniju_indeksas);
                }
                else
                {
                    tarpines_linijos_SS(sender, e, 0, i, 1, 1, liniju_indeksas);
                }
            }

            else if (i < 6)
            {
                Cursor.Position = this.PointToScreen(new Point(250, 0));
                prideti_posisteme_ToolStripMenuItem_Click(sender, e);
                tarpines_linijos_SS(sender, e, 1, i-4, 2, 0, liniju_indeksas);
            }
            else
            {
                Cursor.Position = this.PointToScreen(new Point(450, 0));
                prideti_posisteme_ToolStripMenuItem_Click(sender, e);
            }
            Cursor.Position = a;
        }
        this.Invalidate();
    }
}
}
}

```

## Globalus\_kintamieji.cs

```
namespace FT_Final_v1._0_Smart
{
    class Globalus_kintamieji
    {
        public static string[] posistemiu_pavadinimu_masyvas;
        public static int[] posistemiu_x_koordinaciju_masyvas;
        public static int[] posistemiu_y_koordinaciju_masyvas;
        public static string[] antrasciu_pavadinimu_masyvas;
        public static int[] antrasciu_x_koordinaciju_masyvas;
        public static bool antrastes_veleleve;
        public static int antrastes_tagas;
        public static bool mygtuko_veleleve;
        public static int mygtuko_tagas;
        //Posistemės_dialogas
        public static int[] posistemiu_iejimu_skaicius;
        public static string[,] posistemiu_iejimu_pavadinimu_masyvas;
        public static double[,] posistemiu_iejimu_diapozonu_min_masyvas;
        public static double[,] posistemiu_iejimu_diapozonu_max_masyvas;
        public static int[,] posistemiu_iejimu_narystes_funkciju_sk_masyvas;
        public static string[,] posistemiu_iejimu_narystes_funkciju_pavadinimu_masyvas;
        public static double[,] posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas;
        public static double[,] posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas;
        public static double e = 0.0001;
        public static string[] posistemiu_isejimu_pavadinimu_masyvas;
        public static double[,] posistemiu_isejimu_diapozonu_min_masyvas;
        public static double[,] posistemiu_isejimu_diapozonu_max_masyvas;
        public static int[,] posistemiu_isejimu_narystes_funkciju_sk_masyvas;
        public static string[,] posistemiu_isejimu_narystes_funkciju_pavadinimu_masyvas;
        public static double[,] posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas;
        public static double[,] posistemiu_isejimu_narystes_funkciju_neraiskumu_masyvas;
        public static string[,] posistemiu_isoriniu_duomenu_tekstinis_masyvas;
        public static double[,] posistemiu_isoriniu_duomenu_masyvas;
        public static int[,] posistemiu_taisykliu_skaiciaus_masyvas;
        public static string[,] posistemiu_taisykliu_pavadinimu_masyvas;
        public static string[,] posistemiu_taisykliu_aktyviu_iejimu_indeksu_masyvas;
        public static double[,] posistemiu_taisykliu_vidurkiu_masyvas;
        public static double[,] posistemiu_taisykliu_neraiskumu_masyvas;
        public static double[,] posistemiu_isejimu_taisykliu_vidurkiu_masyvas;
        public static double[,] posistemiu_isejimu_taisykliu_neraiskumu_masyvas;
        public static double[,] posistemiu_rezultatu_masyvas;
        public static string[,] posistemiu_rezultatu_veleveliu_masyvas;
    }
}
```

## Klases.cs

```
using System;
using System.ComponentModel;
using System.Drawing;
using System.Runtime.InteropServices;
using System.Windows.Forms;

namespace FT_Final_v1._0_Smart
{
    class Klases
    {
        public class TestColorTable : ProfessionalColorTable // Naujos spalvų lentelės klasė
        {
            public override Color MenuItemSelected // Pažymėto sub-meniu spalva
            {
                get { return Color.LimeGreen; }
            }
            public override Color ToolStripDropDownBackground // Sub-meniu spalva
            {
                get { return Color.MediumSeaGreen; }
            }
            public override Color MenuItemSelectedGradientBegin // Pažymėto meniu elemento spalva pradžioje
            {
                get { return Color.LimeGreen; }
            }
            public override Color MenuItemPressedGradientEnd // Pažymėto meniu elemento spalva pabaigoje
            {
                get { return Color.LimeGreen; }
            }
        }
    }
}
```



```

public class ReadOnlyTextBox : TextBox
{
    [DllImport("user32.dll")]
    static extern bool HideCaret(IntPtr hWnd);
    public ReadOnlyTextBox()
    {
        this.ReadOnly = true;
        this.GotFocus += TextBoxGotFocus;
        this.Cursor = Cursors.Arrow; // mouse cursor like in other controls
    }
    private void TextBoxGotFocus(object sender, EventArgs args)
    {
        HideCaret(this.Handle);
    }
}
class Line
{
    public Point Start { get; set; }
    public Point End { get; set; }
    public int Width { get; set; }
    public Color Color { get; set; }
}
public class DoubleClickButton : Button
{
    public DoubleClickButton()
    {
        SetStyle(ControlStyles.StandardClick, true);
        SetStyle(ControlStyles.StandardDoubleClick, true);
    }
    [Browsable(true)]
    public new event EventHandler DoubleClick
    {
        add { base.DoubleClick += value; }
        remove { base.DoubleClick -= value; }
    }
}
}

```

---

### Pervadinti.cs

---

```

using System;
using System.Windows.Forms;

namespace FT_Final_v1._0_Smart
{
    public partial class Pervadinti : Form
    {
        public Pervadinti()
        {
            InitializeComponent();
        }

        private void Pervadinti_Load(object sender, EventArgs e)
        {
            if (Globalus_kintamieji.antrastes_velele)
            {
                pervadinti_textBox.Text =
                Globalus_kintamieji.antrasciu_pavadinimu_masyvas[Globalus_kintamieji.antrastes_tagas];
                pervadinti_textBox.MaxLength = 20;
            }
            else if (Globalus_kintamieji.mygtuko_velele)
            {
                pervadinti_textBox.Text =
                Globalus_kintamieji.posistemiu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas];
                pervadinti_textBox.MaxLength = 20;
            }
        }

        private void gerai_button_Click(object sender, EventArgs e)
        {
            if (Globalus_kintamieji.antrastes_velele)
            {
                if (!string.IsNullOrWhiteSpace(pervadinti_textBox.Text))
                {
                    Globalus_kintamieji.antrasciu_pavadinimu_masyvas[Globalus_kintamieji.antrastes_tagas] =
                    pervadinti_textBox.Text;
                }
            }
        }
    }
}

```



```

TextBox isejimo_narystes_funkciju_neraiskumu_textbox = new TextBox();
Label isejimo_narystes_funkciju_pavadinimu_pervadinti_label = new Label();
TextBox isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox = new TextBox();
Chart isejimo_narystes_funkciju_chart = new Chart();
//Taisyklės
Label taisykles_label = new Label();
TextBox taisykles_textbox = new TextBox();
CheckBox taisykles_ir_checkbox = new CheckBox();
CheckBox taisykles_arba_checkbox = new CheckBox();
Button taisykles_isvalyti_pasirinkimus_button = new Button();
Button taisykles_prideti_button = new Button();
Button taisykles_isvalyti_button = new Button();
ListBox[] taisykliu_listbox_masyvas = new ListBox[1];
Label[] taisykliu_antrasciu_masyvas = new Label[1];
//
private void paslepti()
{
    foreach (Control c in this.Controls)
    {
        if (c.Name != iejimai_button.Name && c.Name != narystes_funkcijos_button.Name && c.Name !=
duomenys_button.Name && c.Name != isejimas_button.Name && c.Name != taisykles_button.Name)
        {
            c.Hide();
        }
        else
        {
            c.BackColor = Color.Gainsboro;
        }
    }
}

bool tikrinimo_salyga;
bool pasirinkto_indekso_salyga;
bool pasirinkto_indekso_salyga_funkcijos;
bool pasirinkto_indekso_salyga_isejimas;
private void Posistemes_dialogas_Load(object sender, EventArgs e)
{
    this.Text = Globalus_kintamieji.posistemiui_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas];

//Iėjimai
//
iejimu_sk_label.Text = "Iėjimų skaičius:";
iejimu_sk_label.AutoSize = true;
this.Controls.Add(iejimu_sk_label);
iejimu_sk_label.Location = new Point(atstumas_is_kaires, atstumas_is_virsaus);
//
iejimu_sk_textbox.Text = "Neįvestas";
this.Controls.Add(iejimu_sk_textbox);
iejimu_sk_textbox.Location = new Point(iejimu_sk_label.Location.X, iejimu_sk_label.Location.Y +
iejimu_sk_label.Height+atstumas_is_virsaus_tarp);
iejimu_sk_textbox.Enter +=new EventHandler(iejimu_sk_textbox_aktyvavimo_ivykis);
iejimu_sk_textbox.Validating += new CancelEventHandler(iejimu_sk_textbox_tikrinimo_ivykis);
iejimu_sk_textbox.Validated += new EventHandler(iejimu_sk_textbox_veiksmo_ivykis);
iejimu_sk_textbox.Leave += new EventHandler(iejimu_sk_textbox_deaktyvavimo_ivykis);
//
iejimu_pavadinimu_listbox_label.Text = "Iėjimai:";
iejimu_pavadinimu_listbox_label.AutoSize = true;
this.Controls.Add(iejimu_pavadinimu_listbox_label);
iejimu_pavadinimu_listbox_label.Location = new Point(iejimu_sk_label.Location.X,
iejimu_sk_textbox.Location.Y + iejimu_sk_textbox.Height + atstumas_is_virsaus_tarp);
//
this.Controls.Add(iejimu_pavadinimu_listbox);
iejimu_pavadinimu_listbox.Location = new Point(iejimu_sk_label.Location.X,
iejimu_pavadinimu_listbox_label.Location.Y + iejimu_pavadinimu_listbox_label.Height+atstumas_is_virsaus_tarp);
iejimu_pavadinimu_listbox.Enter += new EventHandler(iejimu_pavadinimu_listbox_aktyvavimo_ivykis);
iejimu_pavadinimu_listbox.Leave += new EventHandler(iejimu_pavadinimu_listbox_deaktyvavimo_ivykis);
iejimu_pavadinimu_listbox.SelectedIndexChanged += new
EventHandler(iejimu_pavadinimu_listbox_pasirinkti_indekso_pasikeitimo_ivykis);
//
iejimu_pavadinimu_pervadinti_label.Text = "Pervadinti";
iejimu_pavadinimu_pervadinti_label.AutoSize = true;
this.Controls.Add(iejimu_pavadinimu_pervadinti_label);
iejimu_pavadinimu_pervadinti_label.Location = new Point(iejimu_sk_label.Location.X,
iejimu_pavadinimu_listbox.Location.Y+ iejimu_pavadinimu_listbox.Height+ atstumas_is_virsaus_tarp);

```

```

//
iejimu_pavadinimu_pervadinti_textbox.Text = "Nepasirinkta";
iejimu_pavadinimu_pervadinti_textbox.MaxLength = 15;
this.Controls.Add(iejimu_pavadinimu_pervadinti_textbox);
iejimu_pavadinimu_pervadinti_textbox.Location = new Point(iejimu_sk_label.Location.X,
iejimu_pavadinimu_pervadinti_label.Location.Y+
iejimu_pavadinimu_pervadinti_label.Height+atstumas_is_virsaus_tarp);
iejimu_pavadinimu_pervadinti_textbox.Enter += new
EventHandler(iejimu_pavadinimu_pervadinti_textbox_akyvavimo_ivykis);
iejimu_pavadinimu_pervadinti_textbox.Validating += new
CancelEventHandler(iejimu_pavadinimu_pervadinti_textbox_tikrinimo_ivykis);
iejimu_pavadinimu_pervadinti_textbox.Validated += new
EventHandler(iejimu_pavadinimu_pervadinti_textbox_veiksmo_ivykis);
iejimu_pavadinimu_pervadinti_textbox.Leave += new
EventHandler(iejimu_pavadinimu_pervadinti_textbox_deaktyvavimo_ivykis);
//
iejimu_diapozono_label.Text = "Diapazonas";
iejimu_diapozono_label.AutoSize = true;
this.Controls.Add(iejimu_diapozono_label);
iejimu_diapozono_label.Location = new Point(2*iejimu_sk_label.Location.X+
iejimu_pavadinimu_listbox.Width, iejimu_pavadinimu_listbox.Location.Y);
//
iejimu_diapozono_min_label.Text = "Min";
iejimu_diapozono_min_label.AutoSize = true;
this.Controls.Add(iejimu_diapozono_min_label);
iejimu_diapozono_min_label.Location = new Point(iejimu_diapozono_label.Location.X,
iejimu_diapozono_label.Location.Y + iejimu_diapozono_label.Height+atstumas_is_virsaus_tarp);
//
iejimu_diapozono_max_label.Text = "Max";
iejimu_diapozono_max_label.AutoSize = true;
this.Controls.Add(iejimu_diapozono_max_label);
iejimu_diapozono_max_label.Location = new Point(iejimu_diapozono_label.Location.X +
iejimu_diapozono_min_textbox.Width+atstumas_is_kaires, iejimu_diapozono_min_label.Location.Y);
//
iejimu_diapozono_min_textbox.Text = "Nepasirinkta";
this.Controls.Add(iejimu_diapozono_min_textbox);
iejimu_diapozono_min_textbox.Location = new Point(iejimu_diapozono_label.Location.X,
iejimu_diapozono_min_label.Location.Y + iejimu_diapozono_min_label.Height+atstumas_is_virsaus_tarp);
iejimu_diapozono_min_textbox.Enter += new
EventHandler(iejimu_diapozono_min_textbox_akyvavimo_ivykis);
iejimu_diapozono_min_textbox.Validating += new
CancelEventHandler(iejimu_diapozono_min_textbox_tikrinimo_ivykis);
iejimu_diapozono_min_textbox.Validated += new
EventHandler(iejimu_diapozono_min_textbox_veiksmo_ivykis);
iejimu_diapozono_min_textbox.Leave += new
EventHandler(iejimu_diapozono_min_textbox_deaktyvavimo_ivykis);
//
iejimu_diapozono_max_textbox.Text = "Nepasirinkta";
this.Controls.Add(iejimu_diapozono_max_textbox);
iejimu_diapozono_max_textbox.Location = new Point(iejimu_diapozono_label.Location.X +
iejimu_diapozono_min_textbox.Width+atstumas_is_kaires, iejimu_diapozono_min_textbox.Location.Y);
iejimu_diapozono_max_textbox.Enter += new
EventHandler(iejimu_diapozono_max_textbox_akyvavimo_ivykis);
iejimu_diapozono_max_textbox.Validating += new
CancelEventHandler(iejimu_diapozono_max_textbox_tikrinimo_ivykis);
iejimu_diapozono_max_textbox.Validated += new
EventHandler(iejimu_diapozono_max_textbox_veiksmo_ivykis);
iejimu_diapozono_max_textbox.Leave += new
EventHandler(iejimu_diapozono_max_textbox_deaktyvavimo_ivykis);
//
iejimu_narystes_funkciju_skaiciaus_label.Text = "Narystės funkcijų skaičius";
iejimu_narystes_funkciju_skaiciaus_label.AutoSize = true;
this.Controls.Add(iejimu_narystes_funkciju_skaiciaus_label);
iejimu_narystes_funkciju_skaiciaus_label.Location = new Point(iejimu_sk_label.Location.X,
iejimu_pavadinimu_pervadinti_textbox.Location.Y + iejimu_pavadinimu_pervadinti_textbox.Height +
atstumas_is_virsaus_tarp);
//
iejimu_narystes_funkciju_skaiciaus_textbox.Text = "Nepasirinkta";
this.Controls.Add(iejimu_narystes_funkciju_skaiciaus_textbox);
iejimu_narystes_funkciju_skaiciaus_textbox.Location = new Point(iejimu_sk_label.Location.X,
iejimu_narystes_funkciju_skaiciaus_label.Location.Y +
iejimu_narystes_funkciju_skaiciaus_label.Height+atstumas_is_virsaus_tarp);
iejimu_narystes_funkciju_skaiciaus_textbox.Enter += new
EventHandler(iejimu_narystes_funkciju_skaiciaus_textbox_akyvavimo_ivykis);
iejimu_narystes_funkciju_skaiciaus_textbox.Validating += new
CancelEventHandler(iejimu_narystes_funkciju_skaiciaus_textbox_tikrinimo_ivykis);

```

```

        iejimu_narystes_funkciju_skaiciaus_textbox.Validated += new
EventHandler(iejimu_narystes_funkciju_skaiciaus_textbox_veiksma_ivykis);
        iejimu_narystes_funkciju_skaiciaus_textbox.Leave += new
EventHandler(iejimu_narystes_funkciju_skaiciaus_textbox_deaktyvavimo_ivykis);
        //
        //Funkcijos
        //
        narystes_funkciju_listbox_label.Text = "Pasirinkto įėjimo narystės funkcijos";
        narystes_funkciju_listbox_label.AutoSize = true;
        this.Controls.Add(narystes_funkciju_listbox_label);
        narystes_funkciju_listbox_label.Location = new Point(atstumas_is_kaires, atstumas_is_virsaus);
        //
        this.Controls.Add(narystes_funkciju_listbox);
        narystes_funkciju_listbox.Location = new Point(narystes_funkciju_listbox_label.Location.X,
narystes_funkciju_listbox_label.Location.Y + narystes_funkciju_listbox_label.Height + atstumas_is_virsaus_tarp);
        narystes_funkciju_listbox.Enter += new EventHandler(narystes_funkciju_listbox_aktyvavimo_ivykis);
        narystes_funkciju_listbox.SelectedIndexChanged += new
EventHandler(narystes_funkciju_listbox_pasirinko_indekso_pasikeitimo_ivykis);
        narystes_funkciju_listbox.Leave += new EventHandler(narystes_funkciju_listbox_deaktyvavimo);
        //
        narystes_funkciju_pervadinti_label.Text = "Pervadinti";
        narystes_funkciju_pervadinti_label.AutoSize = true;
        this.Controls.Add(narystes_funkciju_pervadinti_label);
        narystes_funkciju_pervadinti_label.Location = new Point(2 *
narystes_funkciju_listbox_label.Location.X + narystes_funkciju_listbox.Width,
narystes_funkciju_listbox.Location.Y);
        //
        narystes_funkciju_pervadinti_textbox.Text = "Nepasirinkta";
        narystes_funkciju_pervadinti_textbox.MaxLength = 15;
        this.Controls.Add(narystes_funkciju_pervadinti_textbox);
        narystes_funkciju_pervadinti_textbox.Location = new
Point(narystes_funkciju_pervadinti_label.Location.X, narystes_funkciju_pervadinti_label.Location.Y +
narystes_funkciju_pervadinti_label.Height + atstumas_is_virsaus_tarp);
        narystes_funkciju_pervadinti_textbox.Enter += new
EventHandler(narystes_funkciju_pervadinti_textbox_aktyvavimo_ivykis);
        narystes_funkciju_pervadinti_textbox.Validating += new
CancelEventHandler(narystes_funkciju_pervadinti_textbox_tikrinimo_ivykis);
        narystes_funkciju_pervadinti_textbox.Validated += new
EventHandler(narystes_funkciju_pervadinti_textbox_veiksma_ivykis);
        narystes_funkciju_pervadinti_textbox.Leave += new
EventHandler(narystes_funkciju_pervadinti_textbox_deaktyvavimo_ivykis);
        //
        narystes_funkciju_vidurkiu_label.Text = "Vidurkis";
        narystes_funkciju_vidurkiu_label.AutoSize = true;
        this.Controls.Add(narystes_funkciju_vidurkiu_label);
        narystes_funkciju_vidurkiu_label.Location = new Point(narystes_funkciju_pervadinti_label.Location.X,
narystes_funkciju_pervadinti_textbox.Location.Y + narystes_funkciju_pervadinti_textbox.Height +
atstumas_is_virsaus_tarp);
        //
        narystes_funkciju_vidurkiu_textbox.Text = "Nepasirinkta";
        this.Controls.Add(narystes_funkciju_vidurkiu_textbox);
        narystes_funkciju_vidurkiu_textbox.Location = new
Point(narystes_funkciju_pervadinti_label.Location.X, narystes_funkciju_vidurkiu_label.Location.Y +
narystes_funkciju_vidurkiu_label.Height + atstumas_is_virsaus_tarp);
        narystes_funkciju_vidurkiu_textbox.Enter += new
EventHandler(narystes_funkciju_vidurkiu_textbox_aktyvavimo_ivykis);
        narystes_funkciju_vidurkiu_textbox.Validating += new
CancelEventHandler(narystes_funkciju_vidurkiu_textbox_tikrinimo_ivykis);
        narystes_funkciju_vidurkiu_textbox.Validated += new
EventHandler(narystes_funkciju_vidurkiu_textbox_veiksma_ivykis);
        narystes_funkciju_vidurkiu_textbox.Leave += new
EventHandler(narystes_funkciju_vidurkiu_textbox_deaktyvavimo_ivykis);
        //
        narystes_funkciju_neraiskumu_label.Text = "Neraiskumas";
        narystes_funkciju_neraiskumu_label.AutoSize = true;
        this.Controls.Add(narystes_funkciju_neraiskumu_label);
        narystes_funkciju_neraiskumu_label.Location = new
Point(narystes_funkciju_pervadinti_label.Location.X+
narystes_funkciju_vidurkiu_textbox.Width+atstumas_is_kaires, narystes_funkciju_vidurkiu_label.Location.Y);
        //
        narystes_funkciju_neraiskumu_textbox.Text = "Nepasirinkta";
        this.Controls.Add(narystes_funkciju_neraiskumu_textbox);
        narystes_funkciju_neraiskumu_textbox.Location = new
Point(narystes_funkciju_neraiskumu_label.Location.X, narystes_funkciju_neraiskumu_label.Location.Y +
narystes_funkciju_neraiskumu_label.Height + atstumas_is_virsaus_tarp);
        narystes_funkciju_neraiskumu_textbox.Enter += new
EventHandler(narystes_funkciju_neraiskumu_textbox_aktyvavimo_ivykis);

```

```

        narystes_funkciju_neraiskumu_textbox.Validating += new
CancelEventHandler(narystes_funkciju_neraiskumu_textbox_tikrinimo_ivykis);
        narystes_funkciju_neraiskumu_textbox.Validated += new
EventHandler(narystes_funkciju_neraiskumu_textbox_veiksma_ivykis);
        narystes_funkciju_neraiskumu_textbox.Leave += new
EventHandler(narystes_funkciju_neraiskumu_textbox_deaktyvavimo_ivykis);
        //
        iejimu_narystes_funkciju_chart.Size = new Size(600,250);
        this.Controls.Add(iejimu_narystes_funkciju_chart);
        iejimu_narystes_funkciju_chart.Location = new Point(narystes_funkciju_listbox_label.Location.X,
narystes_funkciju_listbox.Location.Y + narystes_funkciju_listbox.Height + atstumas_is_virsaus_tarp);
        iejimu_narystes_funkciju_chart.MouseDown += new
MouseEventHandler(iejimu_narystes_funkciju_chart_peles_paspaudimo_ivykis);
        iejimu_narystes_funkciju_chart.MouseMove += new
MouseEventHandler(iejimu_narystes_funkciju_chart_peles_judesio_ivykis);
        iejimu_narystes_funkciju_chart.MouseUp += new
MouseEventHandler(iejimu_narystes_funkciju_chart_peles_atleidimo_ivykis);
        //
        //Duomenys
        //
        duomenys_label.Text = "Išoriniai duomenys";
        duomenys_label.AutoSize = true;
        this.Controls.Add(duomenys_label);
        duomenys_label.Location = new Point(atstumas_is_kaires, atstumas_is_virsaus);
        //
        duomenys_textbox.ReadOnly = true;
        duomenys_textbox.Multiline = true;
        duomenys_textbox.ScrollBars = ScrollBars.Both;
        duomenys_textbox.Size = new Size(280, 150);
        this.Controls.Add(duomenys_textbox);
        duomenys_textbox.Location = new Point(duomenys_label.Location.X, duomenys_label.Location.Y +
duomenys_label.Height + atstumas_is_virsaus_tarp);
        //
        duomenys_ieskoti_button.Text = "Pridėti";
        this.Controls.Add(duomenys_ieskoti_button);
        duomenys_ieskoti_button.Location = new Point(duomenys_label.Location.X + duomenys_textbox.Width +
atstumas_is_kaires, duomenys_textbox.Location.Y);
        duomenys_ieskoti_button.Click += new EventHandler(duomenys_ieskoti_button_paspaudimo_ivykis);
        //
        duomenys_isvalyti_button.Text = "Išvalyti";
        this.Controls.Add(duomenys_isvalyti_button);
        duomenys_isvalyti_button.Location = new Point(duomenys_ieskoti_button.Location.X,
duomenys_ieskoti_button.Location.Y + duomenys_ieskoti_button.Height + atstumas_is_virsaus_tarp);
        duomenys_isvalyti_button.Click += new EventHandler(duomenys_isvalyti_button_paspaudimo_ivykis);
        //
        //Išėjimas
        //
        isejimo_label.Text = "Išėjimas:";
        isejimo_label.AutoSize = true;
        this.Controls.Add(isejimo_label);
        isejimo_label.Location = new Point(atstumas_is_kaires, atstumas_is_virsaus);
        //
        if (Globalus_kintamieji.posistemiu_isejimu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas] ==
null)
        {
            Globalus_kintamieji.posistemiu_isejimu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas] =
"Išėjimas" + Globalus_kintamieji.mygtuko_tagas;
            Globalus_kintamieji.posistemiu_isejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas]
= 0;
            Globalus_kintamieji.posistemiu_isejimu_diapozonu_max_masyvas[Globalus_kintamieji.mygtuko_tagas]
= 1;
            Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas] = 0;
        }
        isejimo_textbox.Text =
Globalus_kintamieji.posistemiu_isejimu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas];
        isejimo_textbox.MaxLength = 15;
        this.Controls.Add(isejimo_textbox);
        isejimo_textbox.Location = new Point(isejimo_label.Location.X, isejimo_label.Location.Y +
isejimo_label.Height + atstumas_is_virsaus_tarp);
        isejimo_textbox.Enter += new EventHandler(isejimo_textbox_aktyvavimo_ivykis);
        isejimo_textbox.Validating += new CancelEventHandler(isejimo_textbox_tikrinimo_ivykis);
        isejimo_textbox.Validated += new EventHandler(isejimo_textbox_veiksma_ivykis);
        isejimo_textbox.Leave += new EventHandler(isejimo_textbox_textbox_deaktyvavimo_ivykis);
        //
        isejimo_diapozono_label.Text = "Diapazonas";
        isejimo_diapozono_label.AutoSize = true;

```

```

        this.Controls.Add(isejimo_diapozono_label);
        isejimo_diapozono_label.Location = new Point(isejimo_label.Location.X, isejimo_textbox.Location.Y+
isejimo_textbox.Height+atstumas_is_virsaus_tarp);
        //
        isejimo_diapozono_min_label.Text = "Min";
        isejimo_diapozono_min_label.AutoSize = true;
        this.Controls.Add(isejimo_diapozono_min_label);
        isejimo_diapozono_min_label.Location = new Point(isejimo_label.Location.X,
isejimo_diapozono_label.Location.Y + isejimo_diapozono_label.Height + atstumas_is_virsaus_tarp);
        //
        isejimo_diapozono_max_label.Text = "Max";
        isejimo_diapozono_max_label.AutoSize = true;
        this.Controls.Add(isejimo_diapozono_max_label);
        isejimo_diapozono_max_label.Location = new Point(isejimo_label.Location.X +
isejimo_diapozono_min_textbox.Width + atstumas_is_kaires, isejimo_diapozono_min_label.Location.Y);
        //
        isejimo_diapozono_min_textbox.Text =
Globalus_kintamieji.posistemiu_isejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas].ToString();
        this.Controls.Add(isejimo_diapozono_min_textbox);
        isejimo_diapozono_min_textbox.Location = new Point(isejimo_label.Location.X,
isejimo_diapozono_min_label.Location.Y + isejimo_diapozono_min_label.Height + atstumas_is_virsaus_tarp);
        isejimo_diapozono_min_textbox.Enter += new
EventHandler(isejimo_diapozono_min_textbox_aktyvavimo_ivykis);
        isejimo_diapozono_min_textbox.Validating += new
CancelEventHandler(isejimo_diapozono_min_textbox_tikrinimo_ivykis);
        isejimo_diapozono_min_textbox.Validated += new
EventHandler(isejimo_diapozono_min_textbox_veiksma_ivykis);
        isejimo_diapozono_min_textbox.Leave += new
EventHandler(isejimo_diapozono_min_textbox_deaktyvavimo_ivykis);
        //
        isejimo_diapozono_max_textbox.Text =
Globalus_kintamieji.posistemiu_isejimu_diapozonu_max_masyvas[Globalus_kintamieji.mygtuko_tagas].ToString();
        this.Controls.Add(isejimo_diapozono_max_textbox);
        isejimo_diapozono_max_textbox.Location = new Point(isejimo_diapozono_max_label.Location.X,
isejimo_diapozono_max_label.Location.Y+ isejimo_diapozono_max_label.Height+atstumas_is_virsaus_tarp);
        isejimo_diapozono_max_textbox.Enter += new
EventHandler(isejimo_diapozono_max_textbox_aktyvavimo_ivykis);
        isejimo_diapozono_max_textbox.Validating += new
CancelEventHandler(isejimo_diapozono_max_textbox_tikrinimo_ivykis);
        isejimo_diapozono_max_textbox.Validated += new
EventHandler(isejimo_diapozono_max_textbox_veiksma_ivykis);
        isejimo_diapozono_max_textbox.Leave += new
EventHandler(isejimo_diapozono_max_textbox_deaktyvavimo_ivykis);
        //
        isejimo_narystes_funkciju_skaiciaus_label.Text = "Narystės funkcijų skaičius";
        isejimo_narystes_funkciju_skaiciaus_label.AutoSize = true;
        this.Controls.Add(isejimo_narystes_funkciju_skaiciaus_label);
        isejimo_narystes_funkciju_skaiciaus_label.Location = new Point(isejimo_label.Location.X,
isejimo_diapozono_min_textbox.Location.Y + isejimo_diapozono_min_textbox.Height + atstumas_is_virsaus_tarp);
        //
        isejimo_narystes_funkciju_skaiciaus_textbox.Text =
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas].ToString(
);
        this.Controls.Add(isejimo_narystes_funkciju_skaiciaus_textbox);
        isejimo_narystes_funkciju_skaiciaus_textbox.Location = new Point(isejimo_label.Location.X,
isejimo_narystes_funkciju_skaiciaus_label.Location.Y + isejimo_narystes_funkciju_skaiciaus_label.Height +
atstumas_is_virsaus_tarp);
        isejimo_narystes_funkciju_skaiciaus_textbox.Enter += new
EventHandler(isejimo_narystes_funkciju_skaiciaus_textbox_aktyvavimo_ivykis);
        isejimo_narystes_funkciju_skaiciaus_textbox.Validating += new
CancelEventHandler(isejimo_narystes_funkciju_skaiciaus_textbox_tikrinimo_ivykis);
        isejimo_narystes_funkciju_skaiciaus_textbox.Validated += new
EventHandler(isejimo_narystes_funkciju_skaiciaus_textbox_veiksma_ivykis);
        isejimo_narystes_funkciju_skaiciaus_textbox.Leave += new
EventHandler(isejimo_narystes_funkciju_skaiciaus_textbox_deaktyvavimo_ivykis);
        //
        isejimo_pavadinimu_listbox_label.Text = "Išėjimo narystės funkcijos:";
        isejimo_pavadinimu_listbox_label.AutoSize = true;
        this.Controls.Add(isejimo_pavadinimu_listbox_label);
        isejimo_pavadinimu_listbox_label.Location = new Point(isejimo_label.Location.X + 2 *
isejimo_diapozono_min_textbox.Width + 3 * atstumas_is_virsaus_tarp, isejimo_label.Location.Y);
        //
        this.Controls.Add(isejimo_pavadinimu_listbox);
        isejimo_pavadinimu_listbox.Location = new Point(isejimo_pavadinimu_listbox_label.Location.X,
isejimo_pavadinimu_listbox_label.Location.Y+ isejimo_pavadinimu_listbox_label.Height+atstumas_is_virsaus_tarp);
        isejimo_pavadinimu_listbox.Enter += new EventHandler(isejimo_pavadinimu_listbox_aktyvavimo_ivykis);

```

```

isejimo_pavadinimu_listbox.SelectedIndexChanged += new
EventHandler(isejimo_pavadinimu_listbox_pasirinkto_indeksas_pasikeitimo_ivykis);
isejimo_pavadinimu_listbox.Leave += new
EventHandler(isejimo_pavadinimu_listbox_deaktyvavimo_ivykis);
//
isejimo_narystes_funkciju_vidurkiu_label.Text = "Vidurkis";
isejimo_narystes_funkciju_vidurkiu_label.AutoSize = true;
this.Controls.Add(isejimo_narystes_funkciju_vidurkiu_label);
isejimo_narystes_funkciju_vidurkiu_label.Location = new Point(isejimo_pavadinimu_listbox.Location.X+
isejimo_pavadinimu_listbox.Width+atstumas_is_kaires, isejimo_pavadinimu_listbox.Location.Y);
//
isejimo_narystes_funkciju_vidurkiu_textbox.Text = "Nepasirinkta";
this.Controls.Add(isejimo_narystes_funkciju_vidurkiu_textbox);
isejimo_narystes_funkciju_vidurkiu_textbox.Location = new
Point(isejimo_narystes_funkciju_vidurkiu_label.Location.X, isejimo_narystes_funkciju_vidurkiu_label.Location.Y +
isejimo_narystes_funkciju_vidurkiu_label.Height + atstumas_is_virsaus_tarp);
isejimo_narystes_funkciju_vidurkiu_textbox.Enter += new
EventHandler(isejimo_narystes_funkciju_vidurkiu_textbox_aktyvavimo_ivykis);
isejimo_narystes_funkciju_vidurkiu_textbox.Validating += new
CancelEventHandler(isejimo_narystes_funkciju_vidurkiu_textbox_tikrinimo_ivykis);
isejimo_narystes_funkciju_vidurkiu_textbox.Validated += new
EventHandler(isejimo_narystes_funkciju_vidurkiu_textbox_veiksma_ivykis);
isejimo_narystes_funkciju_vidurkiu_textbox.Leave += new
EventHandler(isejimo_narystes_funkciju_vidurkiu_textbox_deaktyvavimo_ivykis);
//
isejimo_narystes_funkciju_neraiskumu_label.Text = "Neraiškumas";
isejimo_narystes_funkciju_neraiskumu_label.AutoSize = true;
this.Controls.Add(isejimo_narystes_funkciju_neraiskumu_label);
isejimo_narystes_funkciju_neraiskumu_label.Location = new
Point(isejimo_narystes_funkciju_vidurkiu_textbox.Location.X,
isejimo_narystes_funkciju_vidurkiu_textbox.Location.Y+
isejimo_narystes_funkciju_vidurkiu_textbox.Height+atstumas_is_virsaus_tarp);
//
isejimo_narystes_funkciju_neraiskumu_textbox.Text = "Nepasirinkta";
this.Controls.Add(isejimo_narystes_funkciju_neraiskumu_textbox);
isejimo_narystes_funkciju_neraiskumu_textbox.Location = new
Point(isejimo_narystes_funkciju_neraiskumu_label.Location.X,
isejimo_narystes_funkciju_neraiskumu_label.Location.Y + isejimo_narystes_funkciju_neraiskumu_label.Height +
atstumas_is_virsaus_tarp);
isejimo_narystes_funkciju_neraiskumu_textbox.Enter += new
EventHandler(isejimo_narystes_funkciju_neraiskumu_textbox_aktyvavimo_ivykis);
isejimo_narystes_funkciju_neraiskumu_textbox.Validating += new
CancelEventHandler(isejimo_narystes_funkciju_neraiskumu_textbox_tikrinimo_ivykis);
isejimo_narystes_funkciju_neraiskumu_textbox.Validated += new
EventHandler(isejimo_narystes_funkciju_neraiskumu_textbox_veiksma_ivykis);
isejimo_narystes_funkciju_neraiskumu_textbox.Leave += new
EventHandler(isejimo_narystes_funkciju_neraiskumu_textbox_deaktyvavimo_ivykis);
//
isejimo_narystes_funkciju_pavadinimu_pervadinti_label.Text = "Pervadinti";
isejimo_narystes_funkciju_pavadinimu_pervadinti_label.AutoSize = true;
this.Controls.Add(isejimo_narystes_funkciju_pavadinimu_pervadinti_label);
isejimo_narystes_funkciju_pavadinimu_pervadinti_label.Location = new
Point(isejimo_pavadinimu_listbox.Location.X, isejimo_pavadinimu_listbox.Location.Y +
isejimo_pavadinimu_listbox.Height + atstumas_is_virsaus_tarp);
//
isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox.Text = "Nepasirinkta";
isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox.MaxLength = 15;
this.Controls.Add(isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox);
isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox.Location = new
Point(isejimo_narystes_funkciju_pavadinimu_pervadinti_label.Location.X,
isejimo_narystes_funkciju_pavadinimu_pervadinti_label.Location.Y +
isejimo_narystes_funkciju_pavadinimu_pervadinti_label.Height + atstumas_is_virsaus_tarp);
isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox.Enter += new
EventHandler(isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox_aktyvavimo_ivykis);
isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox.Validating += new
CancelEventHandler(isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox_tikrinimo_ivykis);
isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox.Validated += new
EventHandler(isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox_veiksma_ivykis);
isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox.Leave += new
EventHandler(isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox_deaktyvavimo_ivykis);
//
isejimo_narystes_funkciju_chart.Size = new Size(600, 250);
this.Controls.Add(isejimo_narystes_funkciju_chart);
isejimo_narystes_funkciju_chart.Location = new Point(isejimo_label.Location.X,
isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox.Location.Y +
isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox.Height + atstumas_is_virsaus_tarp);

```



```

isejimo_narystes_funkciju_chart.MouseDown += new
MouseEventHandler(isejimo_narystes_funkciju_chart_peles_paspaudimo_ivykis);
isejimo_narystes_funkciju_chart.MouseMove += new
MouseEventHandler(isejimo_narystes_funkciju_chart_peles_judesio_ivykis);
isejimo_narystes_funkciju_chart.MouseUp += new
MouseEventHandler(isejimo_narystes_funkciju_chart_peles_atleidimo_ivykis);
//
//Taisykles
//
taisykles_label.Text = "Taisyklės";
taisykles_label.AutoSize = true;
this.Controls.Add(taisykles_label);
taisykles_label.Location = new Point(atstumas_is_kaires, atstumas_is_virsaus);
//
taisykles_textbox.Multiline = true;
taisykles_textbox.ScrollBars = ScrollBars.Both;
taisykles_textbox.Size = new Size(500,200);
this.Controls.Add(taisykles_textbox);
taisykles_textbox.Location = new Point(taisykles_label.Location.X, taisykles_label.Location.Y +
taisykles_label.Height + atstumas_is_virsaus_tarp);
//
taisykles_ir_checkbox.Text = "Ir";
taisykles_ir_checkbox.AutoSize = true;
this.Controls.Add(taisykles_ir_checkbox);
taisykles_ir_checkbox.Location = new Point(taisykles_label.Location.X, taisykles_textbox.Location.Y
+ taisykles_textbox.Height + atstumas_is_virsaus_tarp);
taisykles_ir_checkbox.CheckedChanged += new
EventHandler(taisykles_ir_checkbox_pazymejimo_pasikeitimo_ivykis);
//
taisykles_arba_checkbox.Text = "Arba";
taisykles_arba_checkbox.AutoSize = true;
this.Controls.Add(taisykles_arba_checkbox);
taisykles_arba_checkbox.Location = new Point(taisykles_label.Location.X +
taisykles_ir_checkbox.Width + atstumas_is_kaires, taisykles_ir_checkbox.Location.Y);
taisykles_arba_checkbox.CheckedChanged += new
EventHandler(taisykles_arba_checkbox_pazymejimo_pasikeitimo_ivykis);
//
taisykles_isvalyti_pasirinkimus_button.Text = "Išvalyti pasirinktus";
taisykles_isvalyti_pasirinkimus_button.AutoSize = true;
this.Controls.Add(taisykles_isvalyti_pasirinkimus_button);
taisykles_isvalyti_pasirinkimus_button.Location = new Point(taisykles_textbox.Location.X +
taisykles_textbox.Width + atstumas_is_kaires, taisykles_textbox.Location.Y);
taisykles_isvalyti_pasirinkimus_button.Click += new
EventHandler(taisykles_isvalyti_pasirinkimus_button_paspaudimo_ivykis);
//
taisykles_prideti_button.Text = "Pridėti";
taisykles_prideti_button.AutoSize = true;
this.Controls.Add(taisykles_prideti_button);
taisykles_prideti_button.Location = new Point(taisykles_isvalyti_pasirinkimus_button.Location.X,
taisykles_isvalyti_pasirinkimus_button.Location.Y+
taisykles_isvalyti_pasirinkimus_button.Height+atstumas_is_virsaus_tarp);
taisykles_prideti_button.Click += new EventHandler(taisykles_prideti_button_paspaudimo_ivykis);
//
taisykles_isvalyti_button.Text = "Išvalyti taisykles";
taisykles_isvalyti_button.AutoSize = true;
this.Controls.Add(taisykles_isvalyti_button);
taisykles_isvalyti_button.Location = new Point(taisykles_isvalyti_pasirinkimus_button.Location.X,
taisykles_prideti_button.Location.Y + taisykles_prideti_button.Height + atstumas_is_virsaus_tarp);
taisykles_isvalyti_button.Click += new EventHandler(taisykles_isvalyti_button_paspaudimo_ivykis);
//
paslepti();
//
//Pakrovimo ivykiai
//
if (Globalus_kintamieji.posistemiu_iejimu_skaicius[Globalus_kintamieji.mygtuko_tagas] > 0)
{
    iejimu_sk_textbox.Text =
Globalus_kintamieji.posistemiu_iejimu_skaicius[Globalus_kintamieji.mygtuko_tagas].ToString();
    iejimu_sk_textbox_aktyvavimo_ivykis(sender, e);
    iejimu_sk_textbox_tikrinimo_ivykis(sender, new CancelEventArgs());
    iejimu_sk_textbox_veiksmo_ivykis(sender, e);
    iejimu_sk_textbox_deaktyvavimo_ivykis(sender, e);
}
if (Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
0] != null)
{
    taisykles_button.PerformClick();
}

```

```

    }
}
//Įėjimai
private void iejimu_sk_textbox_aktyvavimo_ivykis(object sender, EventArgs e)
{
    iejimu_sk_textbox.BackColor = Color.MediumSeaGreen;
}
private void iejimu_sk_textbox_tikrinimo_ivykis(object sender, CancelEventArgs e)
{
    tikrinimo_salyga = false;
    int skaicius;
    if (string.IsNullOrEmpty(iejimu_sk_textbox.Text))
    {
        iejimu_sk_textbox.Text = 0.ToString();
        e.Cancel = false;
        tikrinimo_salyga = false;
    }
    else if (int.TryParse(iejimu_sk_textbox.Text.Trim(), out skaicius))
    {
        if (skaicius >= 0)
        {
            if (skaicius < 100)
            {
                e.Cancel = false;
            }
            else
            {
                MessageBox.Show("Prašau įvesti sveiką skaičių mažesnę nei 100");
                e.Cancel = true;
            }
        }
        else
        {
            MessageBox.Show("Prašau įvesti sveiką skaičių");
            e.Cancel = true;
        }
        tikrinimo_salyga = true;
    }
    else
    {
        MessageBox.Show("Prašau įvesti sveiką skaičių");
        e.Cancel = true;
    }
}
private void iejimu_sk_textbox_veiksma_ivykis(object sender, EventArgs e)
{
    if (tikrinimo_salyga)
    {
        Array.Resize(ref Globalus_kintamieji.posistemiu_iejimu_skaicius,
Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length);
        int.TryParse(iejimu_sk_textbox.Text.Trim(), out
Globalus_kintamieji.posistemiu_iejimu_skaicius[Globalus_kintamieji.mygtuko_tagas]);
        Funkcijos.masyvo_dydzio_keitimas_2d(ref
Globalus_kintamieji.posistemiu_iejimu_pavadinimu_masyvas, Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Lengt
h, Globalus_kintamieji.posistemiu_iejimu_skaicius.Max());
        Funkcijos.masyvo_dydzio_keitimas_2d(ref
Globalus_kintamieji.posistemiu_iejimu_diapozonu_min_masyvas,
Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length, Globalus_kintamieji.posistemiu_iejimu_skaicius.Max());
        Funkcijos.masyvo_dydzio_keitimas_2d(ref
Globalus_kintamieji.posistemiu_iejimu_diapozonu_max_masyvas,
Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length, Globalus_kintamieji.posistemiu_iejimu_skaicius.Max());
        Funkcijos.masyvo_dydzio_keitimas_2d(ref
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_sk_masyvas,
Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length, Globalus_kintamieji.posistemiu_iejimu_skaicius.Max());
        iejimu_pavadinimu_listbox.Items.Clear();
        for (int i = 0; i <
Globalus_kintamieji.posistemiu_iejimu_skaicius[Globalus_kintamieji.mygtuko_tagas]; i++)
        {
            if
(Globalus_kintamieji.posistemiu_iejimu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas, i] == null)
            {
                Globalus_kintamieji.posistemiu_iejimu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas, i] = "Įėjimas"+i;
                Globalus_kintamieji.posistemiu_iejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas, i] = 0;
                Globalus_kintamieji.posistemiu_iejimu_diapozonu_max_masyvas[Globalus_kintamieji.mygtuko_tagas, i] = 1;
            }
        }
    }
}

```

```

Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas, i] = 0;
    }
}

iejimu_pavadinimu_listbox.Items.Add(Globalus_kintamieji.posistemiu_iejimu_pavadinimu_masyvas[Globalus_kintamieji
.mygtuko_tagas, i]);
    }
    iejimu_pavadinimu_listbox_pasirinktas_indeksas = -1;
    iejimu_pavadinimu_listbox.SelectedIndex = -1;
}
}
private void iejimu_sk_textbox_deaktyvavimo_ivykis(object sender, EventArgs e)
{
    iejimu_sk_textbox.BackColor = Color.White;
}
private void iejimu_pavadinimu_listbox_aktyvavimo_ivykis(object sender, EventArgs e)
{
    iejimu_pavadinimu_listbox.BackColor = Color.MediumSeaGreen;
}
private void iejimu_pavadinimu_listbox_deaktyvavimo_ivykis(object sender, EventArgs e)
{
    iejimu_pavadinimu_listbox.BackColor = Color.White;
}
int iejimu_pavadinimu_listbox_pasirinktas_indeksas=-1;
private void iejimu_pavadinimu_listbox_pasirinkti_indekso_pasikeitimo_ivykis(object sender, EventArgs e)
{
    if (iejimu_pavadinimu_listbox.SelectedIndex != -1)
    {
        iejimu_pavadinimu_listbox_pasirinktas_indeksas = iejimu_pavadinimu_listbox.SelectedIndex;
        iejimu_pavadinimu_pervadinti_textbox.Text =
Globalus_kintamieji.posistemiu_iejimu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas];
        iejimu_diapozono_min_textbox.Text=
Globalus_kintamieji.posistemiu_iejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas].ToString();
        iejimu_diapozono_max_textbox.Text =
Globalus_kintamieji.posistemiu_iejimu_diapozonu_max_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas].ToString();
        iejimu_narystes_funkciju_skaiciaus_textbox.Text =
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas].ToString();
    }
}
private void iejimu_pavadinimu_pervadinti_textbox_aktyvavimo_ivykis(object sender, EventArgs e)
{
    pasirinkto_indekso_salyga = false;
    iejimu_pavadinimu_pervadinti_textbox.BackColor = Color.MediumSeaGreen;
    if (iejimu_pavadinimu_listbox_pasirinktas_indeksas == -1)
    {
        MessageBox.Show("Nepasirinktas iðejimas");
        pasirinkto_indekso_salyga = true;
        iejimu_pavadinimu_pervadinti_textbox_tikrinimo_ivykis(sender, new CancelEventArgs());
        iejimu_pavadinimu_pervadinti_textbox_veiksmo_ivykis(sender, e);
        iejimu_diapozono_max_textbox_deaktyvavimo_ivykis(sender, e);
        iejimu_pavadinimu_listbox.Focus();
    }
}
private void iejimu_pavadinimu_pervadinti_textbox_tikrinimo_ivykis(object sender, CancelEventArgs e)
{
    tikrinimo_salyga = false;
    if (!pasirinkto_indekso_salyga)
    {
        if (string.IsNullOrWhiteSpace(iejimu_pavadinimu_pervadinti_textbox.Text))
        {
            iejimu_pavadinimu_pervadinti_textbox.Text =
Globalus_kintamieji.posistemiu_iejimu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas];
            tikrinimo_salyga = false;
        }
        else if (iejimu_pavadinimu_pervadinti_textbox.Text !=
Globalus_kintamieji.posistemiu_iejimu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas])
        {
            tikrinimo_salyga = true;
        }
    }
}
}
}

```

```

    }
    private void iejimu_pavadinimu_pervadinti_textbox_veiksma_ivykis(object sender, EventArgs e)
    {
        if (tikrinimo_salyga)
        {
            Globalus_kintamieji.posistemiu_iejimu_pavadinimu_masyvas[Globalus_kintamieji.myggtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas]= iejimu_pavadinimu_pervadinti_textbox.Text;
            iejimu_pavadinimu_pervadinti_textbox.Text =
Globalus_kintamieji.posistemiu_iejimu_pavadinimu_masyvas[Globalus_kintamieji.myggtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas];
            iejimu_pavadinimu_listbox.Items.Clear();
            for (int i = 0; i <
Globalus_kintamieji.posistemiu_iejimu_skaicius[Globalus_kintamieji.myggtuko_tagas]; i++)
            {
                iejimu_pavadinimu_listbox.Items.Add(Globalus_kintamieji.posistemiu_iejimu_pavadinimu_masyvas[Globalus_kintamieji
.myggtuko_tagas, i]);
            }
            tikrinimo_salyga = false;
        }
    }
    private void iejimu_pavadinimu_pervadinti_textbox_deaktyvavimo_ivykis(object sender, EventArgs e)
    {
        iejimu_pavadinimu_pervadinti_textbox.BackColor = Color.White;
    }
    private void iejimu_diapozono_min_textbox_aktyvavimo_ivykis(object sender, EventArgs e)
    {
        pasirinkto_indekso_salyga = false;
        iejimu_diapozono_min_textbox.BackColor = Color.MediumSeaGreen;
        if (iejimu_pavadinimu_listbox_pasirinktas_indeksas == -1)
        {
            MessageBox.Show("Nepasirinktas įėjimas");
            pasirinkto_indekso_salyga = true;
            iejimu_diapozono_min_textbox_tikrinimo_ivykis(sender, new CancelEventArgs());
            iejimu_diapozono_min_textbox_veiksma_ivykis(sender, e);
            iejimu_diapozono_min_textbox_deaktyvavimo_ivykis(sender, e);
            iejimu_pavadinimu_listbox.Focus();
        }
    }
    private void iejimu_diapozono_min_textbox_tikrinimo_ivykis(object sender, CancelEventArgs e)
    {
        tikrinimo_salyga = false;
        if (!pasirinkto_indekso_salyga)
        {
            tikrinimo_salyga = false;
            double skaicius;
            if (string.IsNullOrWhiteSpace(iejimu_diapozono_min_textbox.Text))
            {
                iejimu_diapozono_min_textbox.Text = 0.ToString();
                e.Cancel = false;
            }
            else if (double.TryParse(iejimu_diapozono_min_textbox.Text.Trim(), out skaicius))
            {
                if (skaicius <
Globalus_kintamieji.posistemiu_iejimu_diapozonu_max_masyvas[Globalus_kintamieji.myggtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas])
                {
                    tikrinimo_salyga = true;
                    e.Cancel = false;
                }
                else
                {
                    MessageBox.Show("Diapozono min reikšmė negali būti didesnė nei max");
                    e.Cancel = true;
                }
            }
        }
    }
    private void iejimu_diapozono_min_textbox_veiksma_ivykis(object sender, EventArgs e)
    {
        if (tikrinimo_salyga)
        {
            Globalus_kintamieji.posistemiu_iejimu_diapozonu_min_masyvas[Globalus_kintamieji.myggtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas] = double.Parse(iejimu_diapozono_min_textbox.Text);
            iejimu_diapozono_min_textbox.Text =
Globalus_kintamieji.posistemiu_iejimu_diapozonu_min_masyvas[Globalus_kintamieji.myggtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas].ToString();
        }
    }
}

```

```

        iejimu_narystes_funkciju_parametru_skaiciavimo_ivykis();
        tikrinimo_salyga = false;
    }
}
private void iejimu_diapozono_min_textbox_deaktyvavimo_ivykis(object sender, EventArgs e)
{
    iejimu_diapozono_min_textbox.BackColor = Color.White;
}
private void iejimu_diapozono_max_textbox_aktyvavimo_ivykis(object sender, EventArgs e)
{
    pasirinkto_indekso_salyga = false;
    iejimu_diapozono_max_textbox.BackColor = Color.MediumSeaGreen;
    if (iejimu_pavadinimu_listbox_pasirinktas_indeksas == -1)
    {
        MessageBox.Show("Nepasirinktas įėjimas");
        pasirinkto_indekso_salyga = true;
        iejimu_diapozono_max_textbox_tikrinimo_ivykis(sender, new CancelEventArgs());
        iejimu_diapozono_max_textbox_veiksmo_ivykis(sender, e);
        iejimu_diapozono_max_textbox_deaktyvavimo_ivykis(sender, e);
        iejimu_pavadinimu_listbox.Focus();
    }
}
private void iejimu_diapozono_max_textbox_tikrinimo_ivykis(object sender, CancelEventArgs e)
{
    tikrinimo_salyga = false;
    if (!pasirinkto_indekso_salyga)
    {
        tikrinimo_salyga = false;
        double skaicius;
        if (string.IsNullOrWhiteSpace(iejimu_diapozono_max_textbox.Text))
        {
            iejimu_diapozono_max_textbox.Text = 0.ToString();
            e.Cancel = false;
        }
        else if (double.TryParse(iejimu_diapozono_max_textbox.Text.Trim(), out skaicius))
        {
            if
(Globalus_kintamieji.posistemiu_iejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuvo_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas] < skaicius)
            {
                tikrinimo_salyga = true;
                e.Cancel = false;
            }
            else
            {
                MessageBox.Show("Diapozono min reikšmė negali būti didesnė nei max");
                e.Cancel = true;
            }
        }
    }
}
private void iejimu_diapozono_max_textbox_veiksmo_ivykis(object sender, EventArgs e)
{
    if (tikrinimo_salyga)
    {
        Globalus_kintamieji.posistemiu_iejimu_diapozonu_max_masyvas[Globalus_kintamieji.mygtuvo_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas] = double.Parse(iejimu_diapozono_max_textbox.Text);
        iejimu_diapozono_max_textbox.Text =
Globalus_kintamieji.posistemiu_iejimu_diapozonu_max_masyvas[Globalus_kintamieji.mygtuvo_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas].ToString();
        iejimu_narystes_funkciju_parametru_skaiciavimo_ivykis();
        tikrinimo_salyga = false;
    }
}
private void iejimu_diapozono_max_textbox_deaktyvavimo_ivykis(object sender, EventArgs e)
{
    iejimu_diapozono_max_textbox.BackColor = Color.White;
}
private void iejimu_narystes_funkciju_skaiciaus_textbox_aktyvavimo_ivykis(object sender, EventArgs e)
{
    pasirinkto_indekso_salyga = false;
    iejimu_narystes_funkciju_skaiciaus_textbox.BackColor = Color.MediumSeaGreen;
    if (iejimu_pavadinimu_listbox_pasirinktas_indeksas == -1)
    {
        MessageBox.Show("Nepasirinktas įėjimas");
        pasirinkto_indekso_salyga = true;
        iejimu_narystes_funkciju_skaiciaus_textbox_tikrinimo_ivykis(sender, new CancelEventArgs());
    }
}

```

```

        iejimu_narystes_funkciju_skaiciaus_textbox_veiksmo_ivykis(sender, e);
        iejimu_narystes_funkciju_skaiciaus_textbox_deaktyvavimo_ivykis(sender, e);
        iejimu_pavadinimu_listbox.Focus();
    }
}
e)
private void iejimu_narystes_funkciju_skaiciaus_textbox_tikrinimo_ivykis(object sender, CancelEventArgs
{
    tikrinimo_salyga = false;
    if (!pasirinkto_indeksas_salyga)
    {
        tikrinimo_salyga = false;
        int skaicius;
        if (string.IsNullOrWhiteSpace(iejimu_narystes_funkciju_skaiciaus_textbox.Text))
        {
            iejimu_narystes_funkciju_skaiciaus_textbox.Text = 0.ToString();
            e.Cancel = false;
            tikrinimo_salyga = false;
        }
        else if (int.TryParse(iejimu_narystes_funkciju_skaiciaus_textbox.Text.Trim(), out skaicius))
        {
            if (skaicius >= 0)
            {
                if (skaicius < 100)
                {
                    e.Cancel = false;
                }
                else
                {
                    MessageBox.Show("Prašau įvesti sveiką skaičių mažesnę nei 100");
                    e.Cancel = true;
                }
            }
            else
            {
                MessageBox.Show("Prašau įvesti sveiką skaičių");
                e.Cancel = true;
            }
            tikrinimo_salyga = true;
        }
        else
        {
            MessageBox.Show("Prašau įvesti sveiką skaičių");
            e.Cancel = true;
        }
    }
}
private void iejimu_narystes_funkciju_parametru_skaiciavimo_ivykis()
{
    for (int i = 0; i <
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas]; i++)
    {
        if
(Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas] == 1) // Reikia apriboti jei = 0
        {
            Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, i] =
(Globalus_kintamieji.posistemiu_iejimu_diapozonu_max_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas] +
Globalus_kintamieji.posistemiu_iejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas]) / 2;

            Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, i] =
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, i] / 3;
        }
        else if
(Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas] == 2)
        {
            if (i == 0)
            {

```



```

iejimu_pavadinimu_listbox_pasirinktas_indeksas] -
Globalus_kintamieji.posistemiu_iejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas]) / 2) / 3;
    }
    }
    else if
(Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas] > 3)
    {
        double size =
(Globalus_kintamieji.posistemiu_iejimu_diapozonu_max_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas] -
Globalus_kintamieji.posistemiu_iejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas]) /
(Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas] - 1);

Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, i] =
Globalus_kintamieji.posistemiu_iejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas] + i * size;

Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, i] =
(Globalus_kintamieji.posistemiu_iejimu_diapozonu_max_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas] -
Globalus_kintamieji.posistemiu_iejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas]) /
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas];
    }
}
}
private void iejimu_narystes_funkciju_skaiciaus_textbox_veiksmo_ivykis(object sender, EventArgs e)
{
    if (tikrinimo_salyga)
    {
        Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas] = int.Parse(iejimu_narystes_funkciju_skaiciaus_textbox.Text);
        iejimu_narystes_funkciju_skaiciaus_textbox.Text =
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas].ToString();
        Funkcijos.masyvo_dydzio_keitimas_3d(ref
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_pavadinimu_masyvas,
Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length, Globalus_kintamieji.posistemiu_iejimu_skaicius.Max(),
Funkcijos.masyvo_antro_stulpelio_max_2d(ref Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_sk_masyvas,
Globalus_kintamieji.mygtuko_tagas));
        Funkcijos.masyvo_dydzio_keitimas_3d(ref
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas,
Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length, Globalus_kintamieji.posistemiu_iejimu_skaicius.Max(),
Funkcijos.masyvo_antro_stulpelio_max_2d(ref Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_sk_masyvas,
Globalus_kintamieji.mygtuko_tagas));
        Funkcijos.masyvo_dydzio_keitimas_3d(ref
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas,
Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length, Globalus_kintamieji.posistemiu_iejimu_skaicius.Max(),
Funkcijos.masyvo_antro_stulpelio_max_2d(ref Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_sk_masyvas,
Globalus_kintamieji.mygtuko_tagas));
        for (int i = 0; i <
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas]; i++)
        {
            if
(Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, i] == null)
            {
                Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, i] = "Narystės funkcija" + i;
            }
        }
        iejimu_narystes_funkciju_parametru_skaiciavimo_ivykis();
        tikrinimo_salyga = false;
    }
}
private void iejimu_narystes_funkciju_skaiciaus_textbox_deaktyvavimo_ivykis(object sender, EventArgs e)
{

```



```

        iejimu_narystes_funkciju_skaiciaus_textbox.BackColor = Color.White;
    }
    //Funkcijos
    private void narystes_funkciju_listbox_aktyvavimo_ivykis(object sender, EventArgs e)
    {
        narystes_funkciju_listbox.BackColor = Color.MediumSeaGreen;
    }
    int narystes_funkciju_listbox_pasirinktas_indeksas=-1;
    private void narystes_funkciju_listbox_pasirinko_indeksas_pasikeitimo_ivykis(object sender, EventArgs e)
    {
        if (narystes_funkciju_listbox.SelectedIndex != -1)
        {
            narystes_funkciju_listbox_pasirinktas_indeksas = narystes_funkciju_listbox.SelectedIndex;
            narystes_funkciju_pervadinti_textbox.Text =
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkciju_listbox_pasirinktas_indeksas];
            narystes_funkciju_vidurkiu_textbox.Text =
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkciju_listbox_pasirinktas_indeksas].ToString();
            narystes_funkciju_neraiskumu_textbox.Text =
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkciju_listbox_pasirinktas_indeksas].ToString();
        }
    }
    private void narystes_funkciju_listbox_deaktyvavimo(object sender, EventArgs e)
    {
        narystes_funkciju_listbox.BackColor = Color.White;
    }
    private void narystes_funkciju_pervadinti_textbox_aktyvavimo_ivykis(object sender, EventArgs e)
    {
        pasirinkto_indeksas_salyga_funkcijos = false;
        narystes_funkciju_pervadinti_textbox.BackColor = Color.MediumSeaGreen;
        if (narystes_funkciju_listbox_pasirinktas_indeksas == -1)
        {
            MessageBox.Show("Nepasirinkta narystės funkcija");
            pasirinkto_indeksas_salyga_funkcijos = true;
            narystes_funkciju_pervadinti_textbox_tikrinimo_ivykis(sender, new CancelEventArgs());
            narystes_funkciju_pervadinti_textbox_veiksma_ivykis(sender, e);
            narystes_funkciju_pervadinti_textbox_deaktyvavimo_ivykis(sender, e);
            narystes_funkciju_listbox.Focus();
        }
    }
    private void narystes_funkciju_pervadinti_textbox_tikrinimo_ivykis(object sender, CancelEventArgs e)
    {
        tikrinimo_salyga = false;
        if (!pasirinkto_indeksas_salyga_funkcijos)
        {
            if (string.IsNullOrWhiteSpace(narystes_funkciju_pervadinti_textbox.Text))
            {
                narystes_funkciju_pervadinti_textbox.Text =
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkciju_listbox_pasirinktas_indeksas];
                tikrinimo_salyga = false;
            }
            else if (narystes_funkciju_pervadinti_textbox.Text !=
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkciju_listbox_pasirinktas_indeksas])
            {
                tikrinimo_salyga = true;
            }
        }
    }
    private void narystes_funkciju_pervadinti_textbox_veiksma_ivykis(object sender, EventArgs e)
    {
        if (tikrinimo_salyga)
        {
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkciju_listbox_pasirinktas_indeksas] =
narystes_funkciju_pervadinti_textbox.Text;
            narystes_funkciju_pervadinti_textbox.Text =
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkciju_listbox_pasirinktas_indeksas];
            narystes_funkciju_listbox.Items.Clear();
            for (int i = 0; i < Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_sk_masyvas
[Globalus_kintamieji.mygtuko_tagas,iejimu_pavadinimu_listbox_pasirinktas_indeksas]; i++)
            {

```

```

narystes_funkciju_listbox.Items.Add(Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_pavadinimu_masyvas[G
lobalus_kintamieji.mygtuko_tagas, iejimu_pavadinimu_listbox_pasirinktas_indeksas, i]);
    }
    iejimu_narystes_funkciju_chart_atnaujinimo_ivykis();
    tikrinimo_salyga = false;
}
}
private void narystes_funkciju_pervadinti_textbox_deaktyvavimo_ivykis(object sender, EventArgs e)
{
    narystes_funkciju_pervadinti_textbox.BackColor = Color.White;
}
private void narystes_funkciju_vidurkiu_textbox_aktyvavimo_ivykis(object sender, EventArgs e)
{
    pasirinkto_indekso_salyga_funkcijos = false;
    narystes_funkciju_vidurkiu_textbox.BackColor = Color.MediumSeaGreen;
    if (narystes_funkciju_listbox_pasirinktas_indeksas == -1)
    {
        MessageBox.Show("Nepasirinkta narystės funkcija");
        pasirinkto_indekso_salyga_funkcijos = true;
        narystes_funkciju_vidurkiu_textbox_tikrinimo_ivykis(sender, new CancelEventArgs());
        narystes_funkciju_vidurkiu_textbox_veiksmo_ivykis(sender, e);
        narystes_funkciju_vidurkiu_textbox_deaktyvavimo_ivykis(sender, e);
        narystes_funkciju_listbox.Focus();
    }
}
private void narystes_funkciju_vidurkiu_textbox_tikrinimo_ivykis(object sender, CancelEventArgs e)
{
    tikrinimo_salyga = false;
    if (!pasirinkto_indekso_salyga_funkcijos)
    {
        tikrinimo_salyga = false;
        double skaicius;
        if (string.IsNullOrEmpty(narystes_funkciju_vidurkiu_textbox.Text))
        {
            narystes_funkciju_vidurkiu_textbox.Text =
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas, iejim
u_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkciju_listbox_pasirinktas_indeksas].ToString();
            e.Cancel = false;
        }
        else if (double.TryParse(narystes_funkciju_vidurkiu_textbox.Text.Trim(), out skaicius))
        {
            if
(Globalus_kintamieji.posistemiu_iejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas] <= skaicius&&
Globalus_kintamieji.posistemiu_iejimu_diapozonu_max_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas]>=skaicius)
            {
                tikrinimo_salyga = true;
                e.Cancel = false;
            }
            else
            {
                MessageBox.Show("Vidurkis turi būti skaičius diapozono ribose");
                e.Cancel = true;
            }
        }
    }
}
private void narystes_funkciju_vidurkiu_textbox_veiksmo_ivykis(object sender, EventArgs e)
{
    if (tikrinimo_salyga)
    {
        Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkciju_listbox_pasirinktas_indeksas] =
double.Parse(narystes_funkciju_vidurkiu_textbox.Text);
        narystes_funkciju_vidurkiu_textbox.Text =
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkciju_listbox_pasirinktas_indeksas].ToString();
        iejimu_narystes_funkciju_chart_atnaujinimo_ivykis();
        tikrinimo_salyga = false;
    }
}
private void narystes_funkciju_vidurkiu_textbox_deaktyvavimo_ivykis(object sender, EventArgs e)
{
    narystes_funkciju_vidurkiu_textbox.BackColor = Color.White;
}

```

```

}
private void narystes_funkciju_neraiskumu_textbox_aktyvavimo_ivykis(object sender, EventArgs e)
{
    pasirinkto_indekso_salyga_funkcijos = false;
    narystes_funkciju_neraiskumu_textbox.BackColor = Color.MediumSeaGreen;
    if (narystes_funkciju_listbox_pasirinktas_indeksas == -1)
    {
        MessageBox.Show("Nepasirinkta narystės funkcija");
        pasirinkto_indekso_salyga_funkcijos = true;
        narystes_funkciju_neraiskumu_textbox_tikrinimo_ivykis(sender, new CancelEventArgs());
        narystes_funkciju_neraiskumu_textbox_veiksma_ivykis(sender, e);
        narystes_funkciju_neraiskumu_textbox_deaktyvavimo_ivykis(sender, e);
        narystes_funkciju_listbox.Focus();
    }
}
private void narystes_funkciju_neraiskumu_textbox_tikrinimo_ivykis(object sender, CancelEventArgs e)
{
    tikrinimo_salyga = false;
    if (!pasirinkto_indekso_salyga_funkcijos)
    {
        tikrinimo_salyga = false;
        double skaicius;
        if (string.IsNullOrEmpty(narystes_funkciju_neraiskumu_textbox.Text))
        {
            narystes_funkciju_neraiskumu_textbox.Text =
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkciju_listbox_pasirinktas_indeksas].ToString();
            e.Cancel = false;
        }
        else if (double.TryParse(narystes_funkciju_neraiskumu_textbox.Text.Trim(), out skaicius))
        {
            if
((Globalus_kintamieji.posistemiu_iejimu_diapozonu_max_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas] -
Globalus_kintamieji.posistemiu_iejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas]) >= skaicius)
            {
                tikrinimo_salyga = true;
                e.Cancel = false;
            }
            else
            {
                MessageBox.Show("Neraiškumas negali būti didesnis nei diapozonas");
                e.Cancel = true;
            }
        }
    }
}
private void narystes_funkciju_neraiskumu_textbox_veiksma_ivykis(object sender, EventArgs e)
{
    if (tikrinimo_salyga)
    {
        Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkciju_listbox_pasirinktas_indeksas] =
double.Parse(narystes_funkciju_neraiskumu_textbox.Text);
        narystes_funkciju_neraiskumu_textbox.Text =
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkciju_listbox_pasirinktas_indeksas].ToString();
        iejimu_narystes_funkciju_chart_atnaujinimo_ivykis();
        tikrinimo_salyga = false;
    }
}
private void narystes_funkciju_neraiskumu_textbox_deaktyvavimo_ivykis(object sender, EventArgs e)
{
    narystes_funkciju_neraiskumu_textbox.BackColor = Color.White;
}
double mastelis;
private void iejimu_narystes_funkciju_chart_atnaujinimo_ivykis()
{
    iejimu_narystes_funkciju_chart.Series.Clear();
    for (int i = 0; i <
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas, iejimu_pava
dinimu_listbox_pasirinktas_indeksas]; i++)
    {
        iejimu_narystes_funkciju_chart.Series.Add("Series" + i);
    }
}

```

```

        iejimu_narystes_funkciju_chart.Series[i].ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Line;
        iejimu_narystes_funkciju_chart.Series[i].LegendText =
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,iejim
imu_pavadinimu_listbox_pasirinktas_indeksas, i];
        mastelis =
(Globalus_kintamieji.posistemiu_iejimu_diapozonu_max_masyvas[Globalus_kintamieji.mygtuko_tagas,iejimu_pavadinimu
_listbox_pasirinktas_indeksas] - Globalus_kintamieji.posistemiu_iejimu_diapozonu_min_masyvas
[Globalus_kintamieji.mygtuko_tagas,iejimu_pavadinimu_listbox_pasirinktas_indeksas]) / 1000;
        for (int j = 0; j < 1000; j++)
        {
            double x =
Globalus_kintamieji.posistemiu_iejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas,iejimu_pavadinimu
_listbox_pasirinktas_indeksas] + j * mastelis;
            double y = Math.Exp(-(x -
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,iejim
u_pavadinimu_listbox_pasirinktas_indeksas, i]) * (x -
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,iejim
u_pavadinimu_listbox_pasirinktas_indeksas, i])) / (2 *
((Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,i
ejimu_pavadinimu_listbox_pasirinktas_indeksas, i]) *
(Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,ie
jimu_pavadinimu_listbox_pasirinktas_indeksas, i]) + Globalus_kintamieji.e));
            iejimu_narystes_funkciju_chart.Series[i].Points.AddXY(x, y);
            iejimu_narystes_funkciju_chart.Series[i].ChartArea = "ChartArea1";
        }
    }
}
int peles_nuspaudimo_vieta1_X;
int peles_nuspaudimo_vieta1_Y;
int peles_nuspaudimo_vieta2_X;
int peles_nuspaudimo_vieta2_Y;
bool narystes_funkcijos_pasirinkimas;
int narystes_funkcijos_indeksas;
private void iejimu_narystes_funkciju_chart_peles_paspaudimo_ivykis(object sender, MouseEventArgs e)
{
    peles_nuspaudimo_vieta1_X=e.X;
    peles_nuspaudimo_vieta1_Y=e.Y;
    for (int i = 0; i <
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas,iejimu_pava
dinimu_listbox_pasirinktas_indeksas]; i++)
    {
        iejimu_narystes_funkciju_chart.Series[i].BorderWidth = 1;
    }
    narystes_funkcijos_pasirinkimas = false;
    HitTestResult kreives_paspaudimas = iejimu_narystes_funkciju_chart.HitTest(e.X,e.Y);
    if (kreives_paspaudimas.ChartElementType == ChartElementType.DataPoint)
    {
        string kreives_pavadinimas = kreives_paspaudimas.Series.Name;
        string kreives_indeksas = Regex.Match(kreives_pavadinimas, @"\d+").Value;
        narystes_funkcijos_indeksas = int.Parse(kreives_indeksas);
        narystes_funkcijos_pasirinkimas = true;
    }
    if (narystes_funkcijos_pasirinkimas)
    {
        iejimu_narystes_funkciju_chart.Series[narystes_funkcijos_indeksas].BorderWidth = 3;
    }
}
int skirtumas;
private void iejimu_narystes_funkciju_chart_peles_judesio_ivykis(object sender, MouseEventArgs e)
{
    peles_nuspaudimo_vieta2_X = e.X;
    peles_nuspaudimo_vieta2_Y = e.Y;

    if (narystes_funkcijos_pasirinkimas)
    {
        if (peles_nuspaudimo_vieta1_X < peles_nuspaudimo_vieta2_X)
        {
            skirtumas = peles_nuspaudimo_vieta2_X - peles_nuspaudimo_vieta1_X;

Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas,narystes_funkcijos_indeksas] =
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkcijos_indeksas] + skirtumas * mastelis * 2;
            narystes_funkciju_vidurkiu_textbox.Text =
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkcijos_indeksas].ToString();

```

```

iejimu_narystes_funkciju_chart.Series[narystes_funkcijos_indeksas].Points.Clear();
for (int j = 0; j < 1000; j++)
{
    double x =
Globalus_kintamieji.posistemiu_iejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas] + j * mastelis;
    double y = Math.Exp(-(x -
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkcijos_indeksas]) * (x -
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkcijos_indeksas])) / (2 *
((Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkcijos_indeksas]) *
(Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkcijos_indeksas])) + Globalus_kintamieji.e));
    iejimu_narystes_funkciju_chart.Series[narystes_funkcijos_indeksas].Points.AddXY(x, y);
}
peles_nuspaudimo_vieta1_X = peles_nuspaudimo_vieta2_X;
}
else if (peles_nuspaudimo_vieta1_X > peles_nuspaudimo_vieta2_X)
{
    skirtumas = peles_nuspaudimo_vieta1_X - peles_nuspaudimo_vieta2_X;

Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkcijos_indeksas] =
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkcijos_indeksas] - skirtumas * mastelis * 2;
    narystes_funkciju_vidurkiu_textbox.Text =
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkcijos_indeksas].ToString();
    iejimu_narystes_funkciju_chart.Series[narystes_funkcijos_indeksas].Points.Clear();
    for (int j = 0; j < 1000; j++)
    {
        double x =
Globalus_kintamieji.posistemiu_iejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas] + j * mastelis;
        double y = Math.Exp(-(x -
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkcijos_indeksas]) * (x -
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkcijos_indeksas])) / (2 *
((Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkcijos_indeksas]) *
(Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkcijos_indeksas])) + Globalus_kintamieji.e));
        iejimu_narystes_funkciju_chart.Series[narystes_funkcijos_indeksas].Points.AddXY(x, y);
    }
    peles_nuspaudimo_vieta1_X = peles_nuspaudimo_vieta2_X;
}
if (peles_nuspaudimo_vieta1_Y < peles_nuspaudimo_vieta2_Y)
{
    skirtumas = peles_nuspaudimo_vieta2_Y - peles_nuspaudimo_vieta1_Y;

Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkcijos_indeksas] =
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkcijos_indeksas] + skirtumas * mastelis * 2;
    narystes_funkciju_neraiskumu_textbox.Text =
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkcijos_indeksas].ToString();
    iejimu_narystes_funkciju_chart.Series[narystes_funkcijos_indeksas].Points.Clear();
    for (int j = 0; j < 1000; j++)
    {
        double x =
Globalus_kintamieji.posistemiu_iejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas] + j * mastelis;
        double y = Math.Exp(-(x -
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkcijos_indeksas]) * (x -
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkcijos_indeksas])) / (2 *
((Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkcijos_indeksas]) *
(Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkcijos_indeksas])) + Globalus_kintamieji.e));
        iejimu_narystes_funkciju_chart.Series[narystes_funkcijos_indeksas].Points.AddXY(x, y);
    }
}

```

```

        peles_nuspaudimo_vieta1_Y = peles_nuspaudimo_vieta2_Y;
    }
    else if (peles_nuspaudimo_vieta1_Y > peles_nuspaudimo_vieta2_Y)
    {
        skirtumas = peles_nuspaudimo_vieta1_Y - peles_nuspaudimo_vieta2_Y;

Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkcijos_indeksas] =
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkcijos_indeksas] - skirtumas * mastelis * 2;
        narystes_funkciju_neraiskumu_textbox.Text =
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkcijos_indeksas].ToString();
        iejimu_narystes_funkciju_chart.Series[narystes_funkcijos_indeksas].Points.Clear();
        for (int j = 0; j < 1000; j++)
        {
            double x =
Globalus_kintamieji.posistemiu_iejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas] + j * mastelis;
            double y = Math.Exp(-(x -
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkcijos_indeksas]) * (x -
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkcijos_indeksas])) / (2 *
((Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkcijos_indeksas]) *
(Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas, narystes_funkcijos_indeksas])) + Globalus_kintamieji.e));
            iejimu_narystes_funkciju_chart.Series[narystes_funkcijos_indeksas].Points.AddXY(x, y);
        }
        peles_nuspaudimo_vieta1_Y = peles_nuspaudimo_vieta2_Y;
    }
}
}
private void iejimu_narystes_funkciju_chart_peles_atleidimo_ivykis(object sender, MouseEventArgs e)
{
    narystes_funkcijos_pasirinkimas = false;
}
//Duomenys
private void duomenys_ieskoti_button_paspaudimo_ivykis(object sender, EventArgs e)
{
    int size = -1;
    var paieskos_dialogas = new OpenFileDialog();
    paieskos_dialogas.Title = "Atidaryti išorinių duomenų.txt";
    paieskos_dialogas.Filter = "Text files (*.txt)|*.txt";
    DialogResult result = paieskos_dialogas.ShowDialog(); // Show the dialog.
    if (result == DialogResult.OK) // Test result.
    {
        string file = paieskos_dialogas.FileName;
        try
        {
            string text = File.ReadAllText(file);

Globalus_kintamieji.posistemiu_isoriniu_duomenu_tekstinis_masyvas[Globalus_kintamieji.mygtuko_tagas] = text;
            size = text.Length;
            duomenys_textbox.Clear();
            duomenys_textbox.Text = text;
            int i=0, j = 0;
            foreach (var eilute in text.Split('\n'))
            {
                j = 0;
                foreach (var stulpelis in eilute.Trim().Split(' '))
                {
                    j++;
                }
                i++;
            }
            Funkcijos.masyvo_dydzio_keitimas_3d(ref
Globalus_kintamieji.posistemiu_isoriniu_duomenu_masyvas,
Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length, i, j);
            i = 0; j = 0;
            foreach (var eilute in text.Split('\n'))
            {
                j = 0;
                foreach (var stulpelis in eilute.Trim().Split(' '))
                {

```

```

Globalus_kintamieji.posistemiu_isoriniu_duomenu_masyvas[Globalus_kintamieji.mygtuko_tagas, i, j] =
Convert.ToDouble(stulpelis.Trim());
        j++;
    }
    i++;
}
}
catch (IOException)
{
    MessageBox.Show("Failas nenuskaitytas, patikrinkite failą");
}
}
Console.WriteLine(size); // <-- Shows file size in debugging mode.
Console.WriteLine(result); // <-- For debugging use.
}
private void duomenys_isvalyti_button_paspaudimo_ivykis(object sender, EventArgs e)
{
    duomenys_textbox.Clear();
    Globalus_kintamieji.posistemiu_isoriniu_duomenu_tekstinis_masyvas[Globalus_kintamieji.mygtuko_tagas]
= null;
}
//Išėjimas
private void isejimo_textbox_aktyvavimo_ivykis(object sender, EventArgs e)
{
    isejimo_textbox.BackColor = Color.MediumSeaGreen;
}
private void isejimo_textbox_tikrinimo_ivykis(object sender, CancelEventArgs e)
{
    tikrinimo_salyga = false;
    if (string.IsNullOrEmpty(isejimo_textbox.Text))
    {
        isejimo_textbox.Text =
Globalus_kintamieji.posistemiu_isejimu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas];
        tikrinimo_salyga = false;
    }
    else if (isejimo_textbox.Text !=
Globalus_kintamieji.posistemiu_isejimu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas])
    {
        tikrinimo_salyga = true;
    }
}
private void isejimo_textbox_veiksma_ivykis(object sender, EventArgs e)
{
    if (tikrinimo_salyga)
    {
        Globalus_kintamieji.posistemiu_isejimu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas] =
isejimo_textbox.Text;
        isejimo_textbox.Text =
Globalus_kintamieji.posistemiu_isejimu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas];
        if
(Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas] > 0)
        {
            isejimo_narystes_funkciju_chart_atnaujinimo_ivykis();
        }
        tikrinimo_salyga = false;
    }
}
private void isejimo_textbox_textbox_deaktyvavimo_ivykis(object sender, EventArgs e)
{
    isejimo_textbox.BackColor = Color.White;
}
private void isejimo_diapozono_min_textbox_aktyvavimo_ivykis(object sender, EventArgs e)
{
    isejimo_diapozono_min_textbox.BackColor = Color.MediumSeaGreen;
}
private void isejimo_diapozono_min_textbox_tikrinimo_ivykis(object sender, CancelEventArgs e)
{
    tikrinimo_salyga = false;
    double skaicius;
    if (isejimo_diapozono_min_textbox.Text ==
Globalus_kintamieji.posistemiu_isejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas].ToString())
    {
        tikrinimo_salyga = false;
        e.Cancel = false;
    }
    else if (string.IsNullOrEmpty(isejimo_diapozono_min_textbox.Text))

```

```

        {
            isejimo_diapozono_min_textbox.Text =
Globalus_kintamieji.posistemiu_isejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas].ToString();
            e.Cancel = false;
        }
        else if (double.TryParse(isejimo_diapozono_min_textbox.Text.Trim(), out skaicius))
        {
            if (skaicius <
Globalus_kintamieji.posistemiu_isejimu_diapozonu_max_masyvas[Globalus_kintamieji.mygtuko_tagas])
            {
                tikrinimo_salyga = true;
                e.Cancel = false;
            }
            else
            {
                MessageBox.Show("Diapazono min reikšmė negali būti didesnė nei max");
                e.Cancel = true;
            }
        }
    }
    private void isejimo_diapozono_min_textbox_veiksma_ivykis(object sender, EventArgs e)
    {
        if (tikrinimo_salyga)
        {
            Globalus_kintamieji.posistemiu_isejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas]
= double.Parse(isejimo_diapozono_min_textbox.Text);
            isejimo_diapozono_min_textbox.Text =
Globalus_kintamieji.posistemiu_isejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas].ToString();
            if
(Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas] > 0)
            {
                isejimu_narystes_funkciju_parametru_skaiciavimo_ivykis();
                isejimu_narystes_funkciju_chart_atnaujinimo_ivykis();
            }
            tikrinimo_salyga = false;
        }
    }
    private void isejimo_diapozono_min_textbox_deaktyvavimo_ivykis(object sender, EventArgs e)
    {
        isejimo_diapozono_min_textbox.BackColor = Color.White;
    }
    private void isejimo_diapozono_max_textbox_aktyvavimo_ivykis(object sender, EventArgs e)
    {
        isejimo_diapozono_max_textbox.BackColor = Color.MediumSeaGreen;
    }
    private void isejimo_diapozono_max_textbox_tikrinimo_ivykis(object sender, CancelEventArgs e)
    {
        tikrinimo_salyga = false;
        double skaicius;
        if (isejimo_diapozono_max_textbox.Text ==
Globalus_kintamieji.posistemiu_isejimu_diapozonu_max_masyvas[Globalus_kintamieji.mygtuko_tagas].ToString())
        {
            tikrinimo_salyga = false;
            e.Cancel = false;
        }
        else if (string.IsNullOrEmpty(isejimo_diapozono_max_textbox.Text))
        {
            isejimo_diapozono_max_textbox.Text =
Globalus_kintamieji.posistemiu_isejimu_diapozonu_max_masyvas[Globalus_kintamieji.mygtuko_tagas].ToString();
            e.Cancel = false;
        }
        else if (double.TryParse(isejimo_diapozono_max_textbox.Text.Trim(), out skaicius))
        {
            if (skaicius >
Globalus_kintamieji.posistemiu_isejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas])
            {
                tikrinimo_salyga = true;
                e.Cancel = false;
            }
            else
            {
                MessageBox.Show("Diapazono min reikšmė negali būti didesnė nei max");
                e.Cancel = true;
            }
        }
    }
}
private void isejimo_diapozono_max_textbox_veiksma_ivykis(object sender, EventArgs e)

```



```

    {
        if (tikrinimo_salyga)
        {
            Globalus_kintamieji.posistemiu_isejimu_diapozonu_max_masyvas[Globalus_kintamieji.mygtuko_tagas]
= double.Parse(isejimo_diapozono_max_textbox.Text);
            isejimo_diapozono_max_textbox.Text =
Globalus_kintamieji.posistemiu_isejimu_diapozonu_max_masyvas[Globalus_kintamieji.mygtuko_tagas].ToString();
            if
(Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas] > 0)
            {
                isejimu_narystes_funkciju_parametru_skaiciavimo_ivykis();
                isejimo_narystes_funkciju_chart_atnaujinimo_ivykis();
            }
            tikrinimo_salyga = false;
        }
    }
    private void isejimo_diapozono_max_textbox_deaktyvavimo_ivykis(object sender, EventArgs e)
    {
        isejimo_diapozono_max_textbox.BackColor = Color.White;
    }
    private void isejimo_narystes_funkciju_skaiciaus_textbox_aktyvavimo_ivykis(object sender, EventArgs e)
    {
        isejimo_narystes_funkciju_skaiciaus_textbox.BackColor = Color.MediumSeaGreen;
    }
    private void isejimo_narystes_funkciju_skaiciaus_textbox_tikrinimo_ivykis(object sender, CancelEventArgs
e)
    {
        tikrinimo_salyga = false;
        int skaicius;
        if (isejimo_narystes_funkciju_skaiciaus_textbox.Text ==
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas].ToString(
))
        {
            tikrinimo_salyga = false;
            e.Cancel = false;
        }
        else if (string.IsNullOrEmpty(isejimo_narystes_funkciju_skaiciaus_textbox.Text))
        {
            isejimo_narystes_funkciju_skaiciaus_textbox.Text =
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas].ToString(
);
            e.Cancel = false;
            tikrinimo_salyga = false;
        }
        else if (int.TryParse(isejimo_narystes_funkciju_skaiciaus_textbox.Text.Trim(), out skaicius))
        {
            if (skaicius >= 0)
            {
                if (skaicius < 100)
                {
                    e.Cancel = false;
                }
                else
                {
                    MessageBox.Show("Prašau įvesti sveiką skaičių mažesnę nei 100");
                    e.Cancel = true;
                }
            }
            else
            {
                MessageBox.Show("Prašau įvesti sveiką skaičių");
                e.Cancel = true;
            }
            tikrinimo_salyga = true;
        }
        else
        {
            MessageBox.Show("Prašau įvesti sveiką skaičių");
            e.Cancel = true;
        }
    }
    private void isejimu_narystes_funkciju_parametru_skaiciavimo_ivykis()
    {
        for (int i = 0; i <
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas]; i++)
        {

```



```

        double size =
(Globalus_kintamieji.posistemiu_isejimu_diapozonu_max_masyvas[Globalus_kintamieji.mygtuko_tagas] -
Globalus_kintamieji.posistemiu_isejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas]) /
(Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas] - 1);

Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas, i]
= Globalus_kintamieji.posistemiu_isejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas] + i * size;

Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
i] = (Globalus_kintamieji.posistemiu_isejimu_diapozonu_max_masyvas[Globalus_kintamieji.mygtuko_tagas] -
Globalus_kintamieji.posistemiu_isejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas]) /
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas];
    }
}
}
private void isejimo_narystes_funkciju_skaiciaus_textbox_veiksma_ivykis(object sender, EventArgs e)
{
    if (tikrinimo_salyga)
    {
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas] =
int.Parse(isejimo_narystes_funkciju_skaiciaus_textbox.Text);
isejimo_narystes_funkciju_skaiciaus_textbox.Text =
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas].ToString(
);
        Funkcijos.masyvo_dydzio_keitimas_2d(ref
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_pavadinimu_masyvas,
Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length,
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_sk_masyvas.Max());
        Funkcijos.masyvo_dydzio_keitimas_2d(ref
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas,
Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length,
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_sk_masyvas.Max());
        Funkcijos.masyvo_dydzio_keitimas_2d(ref
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraiskumu_masyvas,
Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length,
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_sk_masyvas.Max());
isejimo_pavadinimu_listbox.Items.Clear();
        for (int i = 0; i <
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas]; i++)
        {
            if
(Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
i] == null)
            {
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
i] = "Narystės funkcija" + i;
            }
isejimo_pavadinimu_listbox.Items.Add(Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_pavadinimu_masyvas
[Globalus_kintamieji.mygtuko_tagas, i]);
        }
isejimo_narystes_funkciju_parametru_skaiciavimo_ivykis();
isejimo_narystes_funkciju_chart_atnaujinimo_ivykis();
tikrinimo_salyga = false;
    }
}
private void isejimo_narystes_funkciju_skaiciaus_textbox_deaktyvavimo_ivykis(object sender, EventArgs e)
{
    isejimo_narystes_funkciju_skaiciaus_textbox.BackColor = Color.White;
isejimo_pavadinimu_listbox_pasirinktas_indeksas = -1;
isejimo_pavadinimu_listbox.SelectedIndex = -1;
}
private void isejimo_pavadinimu_listbox_aktyvavimo_ivykis(object sender, EventArgs e)
{
    isejimo_pavadinimu_listbox.BackColor = Color.MediumSeaGreen;
}
int isejimo_pavadinimu_listbox_pasirinktas_indeksas=-1;
private void isejimo_pavadinimu_listbox_pasirinkto_indeksas_pasikeitimo_ivykis(object sender, EventArgs
e)
{
    if (isejimo_pavadinimu_listbox.SelectedIndex != -1)
    {
        isejimo_pavadinimu_listbox_pasirinktas_indeksas = isejimo_pavadinimu_listbox.SelectedIndex;
    }
}

```

```

isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox.Text =
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
isejimo_pavadinimu_listbox_pasirinktas_indeksas];
isejimo_narystes_funkciju_vidurkiu_textbox.Text =
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
isejimo_pavadinimu_listbox_pasirinktas_indeksas].ToString();
isejimo_narystes_funkciju_neraiskumu_textbox.Text =
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
isejimo_pavadinimu_listbox_pasirinktas_indeksas].ToString();
isejimo_narystes_funkciju_chart_atnaujinimo_ivykis();
}
}
private void isejimo_pavadinimu_listbox_deaktyvavimo_ivykis(object sender, EventArgs e)
{
isejimo_pavadinimu_listbox.BackColor = Color.White;
}
private void isejimo_narystes_funkciju_vidurkiu_textbox_aktyvavimo_ivykis(object sender, EventArgs e)
{
pasirinkto_indekso_salyga_isejimas = false;
isejimo_narystes_funkciju_vidurkiu_textbox.BackColor = Color.MediumSeaGreen;
if (isejimo_pavadinimu_listbox_pasirinktas_indeksas == -1)
{
MessageBox.Show("Nepasirinkta narystės funkcija");
pasirinkto_indekso_salyga_isejimas = true;
isejimo_narystes_funkciju_vidurkiu_textbox_tikrinimo_ivykis(sender, new CancelEventArgs());
isejimo_narystes_funkciju_vidurkiu_textbox_veiksma_ivykis(sender, e);
isejimo_narystes_funkciju_vidurkiu_textbox_deaktyvavimo_ivykis(sender, e);
isejimo_pavadinimu_listbox.Focus();
}
}
private void isejimo_narystes_funkciju_vidurkiu_textbox_tikrinimo_ivykis(object sender, CancelEventArgs
e)
{
tikrinimo_salyga = false;
if (!pasirinkto_indekso_salyga_isejimas)
{
tikrinimo_salyga = false;
double skaicius;
if (string.IsNullOrEmpty(isejimo_narystes_funkciju_vidurkiu_textbox.Text))
{
isejimo_narystes_funkciju_vidurkiu_textbox.Text =
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
isejimo_pavadinimu_listbox_pasirinktas_indeksas].ToString();
e.Cancel = false;
}
else if (double.TryParse(isejimo_narystes_funkciju_vidurkiu_textbox.Text.Trim(), out skaicius))
{
if
(Globalus_kintamieji.posistemiu_isejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas] <= skaicius &&
Globalus_kintamieji.posistemiu_isejimu_diapozonu_max_masyvas[Globalus_kintamieji.mygtuko_tagas] >= skaicius)
{
tikrinimo_salyga = true;
e.Cancel = false;
}
else
{
MessageBox.Show("Vidurkis turi būti skaičius diapozono ribose");
e.Cancel = true;
}
}
}
}
private void isejimo_narystes_funkciju_vidurkiu_textbox_veiksma_ivykis(object sender, EventArgs e)
{
if (tikrinimo_salyga)
{
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
isejimo_pavadinimu_listbox_pasirinktas_indeksas] =
double.Parse(isejimo_narystes_funkciju_vidurkiu_textbox.Text);
isejimo_narystes_funkciju_vidurkiu_textbox.Text =
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
isejimo_pavadinimu_listbox_pasirinktas_indeksas].ToString();
isejimo_narystes_funkciju_chart_atnaujinimo_ivykis();
tikrinimo_salyga = false;
}
}
}
}

```

```

private void isejimo_narystes_funkciju_vidurkiu_textbox_deaktyvavimo_ivykis(object sender, EventArgs e)
{
    isejimo_narystes_funkciju_vidurkiu_textbox.BackColor = Color.White;
}
private void isejimo_narystes_funkciju_neraiskumu_textbox_aktyvavimo_ivykis(object sender, EventArgs e)
{
    pasirinkto_indekso_salyga_isejimas = false;
    isejimo_narystes_funkciju_neraiskumu_textbox.BackColor = Color.MediumSeaGreen;
    if (isejimo_pavadinimu_listbox_pasirinktas_indeksas == -1)
    {
        MessageBox.Show("Nepasirinkta narystės funkcija");
        pasirinkto_indekso_salyga_isejimas = true;
        isejimo_narystes_funkciju_neraiskumu_textbox_tikrinimo_ivykis(sender, new CancelEventArgs());
        isejimo_narystes_funkciju_neraiskumu_textbox_veiksma_ivykis(sender, e);
        isejimo_narystes_funkciju_neraiskumu_textbox_deaktyvavimo_ivykis(sender, e);
        isejimo_pavadinimu_listbox.Focus();
    }
}
private void isejimo_narystes_funkciju_neraiskumu_textbox_tikrinimo_ivykis(object sender,
CancelEventArgs e)
{
    tikrinimo_salyga = false;
    if (!pasirinkto_indekso_salyga_isejimas)
    {
        tikrinimo_salyga = false;
        double skaicius;
        if (string.IsNullOrEmpty(isejimo_narystes_funkciju_neraiskumu_textbox.Text))
        {
            isejimo_narystes_funkciju_neraiskumu_textbox.Text =
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
isejimo_pavadinimu_listbox_pasirinktas_indeksas].ToString();
            e.Cancel = false;
        }
        else if (double.TryParse(isejimo_narystes_funkciju_neraiskumu_textbox.Text.Trim(), out
skaicius))
        {
            if
((Globalus_kintamieji.posistemiu_isejimu_diapozonu_max_masyvas[Globalus_kintamieji.mygtuko_tagas] -
Globalus_kintamieji.posistemiu_isejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas]) >= skaicius)
            {
                tikrinimo_salyga = true;
                e.Cancel = false;
            }
            else
            {
                MessageBox.Show("Neraiškumas negali būti didesnis nei diapozonas");
                e.Cancel = true;
            }
        }
    }
}
private void isejimo_narystes_funkciju_neraiskumu_textbox_veiksma_ivykis(object sender, EventArgs e)
{
    if (tikrinimo_salyga)
    {
        Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
isejimo_pavadinimu_listbox_pasirinktas_indeksas] =
double.Parse(isejimo_narystes_funkciju_neraiskumu_textbox.Text);
        isejimo_narystes_funkciju_neraiskumu_textbox.Text =
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
isejimo_pavadinimu_listbox_pasirinktas_indeksas].ToString();
        isejimo_narystes_funkciju_chart_atnaujinimo_ivykis();
        tikrinimo_salyga = false;
    }
}
private void isejimo_narystes_funkciju_neraiskumu_textbox_deaktyvavimo_ivykis(object sender, EventArgs
e)
{
    isejimo_narystes_funkciju_neraiskumu_textbox.BackColor = Color.White;
}
private void isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox_aktyvavimo_ivykis(object sender,
EventArgs e)
{
    pasirinkto_indekso_salyga_isejimas = false;
    isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox.BackColor = Color.MediumSeaGreen;
    if (isejimo_pavadinimu_listbox_pasirinktas_indeksas == -1)

```

```

    {
        MessageBox.Show("Nepasirinkta narystės funkcija");
        pasirinkto_indekso_salyga_isejimas = true;
        isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox_tikrinimo_ivykis(sender, new
CancelEventArgs());
        isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox_veiksmo_ivykis(sender, e);
        isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox_deaktyvavimo_ivykis(sender, e);
        isejimo_pavadinimu_listbox.Focus();
    }
}
private void isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox_tikrinimo_ivykis(object sender,
CancelEventArgs e)
{
    tikrinimo_salyga = false;
    if (!pasirinkto_indekso_salyga_isejimas)
    {
        if (string.IsNullOrWhiteSpace(isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox.Text))
        {
            isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox.Text =
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
isejimo_pavadinimu_listbox_pasirinktas_indeksas];
            tikrinimo_salyga = false;
        }
        else if (isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox.Text !=
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
isejimo_pavadinimu_listbox_pasirinktas_indeksas])
        {
            tikrinimo_salyga = true;
        }
    }
}
private void isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox_veiksmo_ivykis(object sender,
EventArgs e)
{
    if (tikrinimo_salyga)
    {
        Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
isejimo_pavadinimu_listbox_pasirinktas_indeksas] = isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox.Text;
        isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox.Text =
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
isejimo_pavadinimu_listbox_pasirinktas_indeksas];
        isejimo_pavadinimu_listbox.Items.Clear();
        for (int i = 0; i <
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas]; i++)
        {
            isejimo_pavadinimu_listbox.Items.Add(Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_pavadinimu_masyvas
[Globalus_kintamieji.mygtuko_tagas, i]);
        }
        isejimo_narystes_funkciju_chart_atnaujinimo_ivykis();
        tikrinimo_salyga = false;
    }
}
private void isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox_deaktyvavimo_ivykis(object sender,
EventArgs e)
{
    isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox.BackColor = Color.White;
}
private void isejimo_narystes_funkciju_chart_atnaujinimo_ivykis()
{
    isejimo_narystes_funkciju_chart.ChartAreas.Clear();
    var chartarea = new ChartArea();
    chartarea.Name = "ChartArea1";
    isejimo_narystes_funkciju_chart.ChartAreas.Add(chartarea);
    isejimo_narystes_funkciju_chart.Titles.Clear();
    var charttitle = new Title();
    charttitle.Name = "Title1";
    isejimo_narystes_funkciju_chart.Legends.Clear();
    var chartlegend = new Legend();
    isejimo_narystes_funkciju_chart.Legends.Add(chartlegend);
    isejimo_narystes_funkciju_chart.Titles.Add(charttitle);
    isejimo_narystes_funkciju_chart.ChartAreas["ChartArea1"].AxisX.Minimum =
Globalus_kintamieji.posistemiu_isejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas];
    isejimo_narystes_funkciju_chart.ChartAreas["ChartArea1"].AxisX.Maximum =
Globalus_kintamieji.posistemiu_isejimu_diapozonu_max_masyvas[Globalus_kintamieji.mygtuko_tagas];
    isejimo_narystes_funkciju_chart.ChartAreas["ChartArea1"].AxisY.Minimum = 0;
}

```

```

isejimo_narystes_funkciju_chart.ChartAreas["ChartArea1"].AxisY.Maximum = 1;
isejimo_narystes_funkciju_chart.Titles["Title1"].Text =
Globalus_kintamieji.posistemiu_isejimu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas];
isejimo_narystes_funkciju_chart.Series.Clear();
for (int i = 0; i <
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas]; i++)
{
    isejimo_narystes_funkciju_chart.Series.Add("Series" + i);
    isejimo_narystes_funkciju_chart.Series[i].ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Line;
    isejimo_narystes_funkciju_chart.Series[i].LegendText =
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
i];
    mastelis =
(Globalus_kintamieji.posistemiu_isejimu_diapozonu_max_masyvas[Globalus_kintamieji.mygtuko_tagas] -
Globalus_kintamieji.posistemiu_isejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas]) / 1000;
    for (int j = 0; j < 1000; j++)
    {
        double x =
Globalus_kintamieji.posistemiu_isejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas] + j * mastelis;
        double y = Math.Exp(-(x -
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas, i])
* (x -
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
i])) / (2 *
((Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
i]) *
(Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
i])) + Globalus_kintamieji.e));
        isejimo_narystes_funkciju_chart.Series[i].Points.AddXY(x, y);
        isejimo_narystes_funkciju_chart.Series[i].ChartArea = "ChartArea1";
    }
}
}
private void isejimo_narystes_funkciju_chart_peles_paspaudimo_ivykis(object sender, MouseEventArgs e)
{
    peles_nuspaudimo_vieta1_X = e.X;
    peles_nuspaudimo_vieta1_Y = e.Y;
    for (int i = 0; i <
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas]; i++)
    {
        isejimo_narystes_funkciju_chart.Series[i].BorderWidth = 1;
    }
    narystes_funkcijos_pasirinkimas = false;
    HitTestResult kreives_paspaudimas = isejimo_narystes_funkciju_chart.HitTest(e.X, e.Y);
    if (kreives_paspaudimas.ChartElementType == ChartElementType.DataPoint)
    {
        string kreives_pavadinimas = kreives_paspaudimas.Series.Name;
        string kreives_indeksas = Regex.Match(kreives_pavadinimas, @"\d+").Value;
        narystes_funkcijos_indeksas = int.Parse(kreives_indeksas);
        narystes_funkcijos_pasirinkimas = true;
    }
    if (narystes_funkcijos_pasirinkimas)
    {
        isejimo_narystes_funkciju_chart.Series[narystes_funkcijos_indeksas].BorderWidth = 3;
    }
}
private void isejimo_narystes_funkciju_chart_peles_judesio_ivykis(object sender, MouseEventArgs e)
{
    peles_nuspaudimo_vieta2_X = e.X;
    peles_nuspaudimo_vieta2_Y = e.Y;

    if (narystes_funkcijos_pasirinkimas)
    {
        if (peles_nuspaudimo_vieta1_X < peles_nuspaudimo_vieta2_X)
        {
            skirtumas = peles_nuspaudimo_vieta2_X - peles_nuspaudimo_vieta1_X;

Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas] =
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas] + skirtumas * mastelis * 2;
            isejimo_narystes_funkciju_vidurkiu_textbox.Text =
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas].ToString();
            isejimo_narystes_funkciju_chart.Series[narystes_funkcijos_indeksas].Points.Clear();
            for (int j = 0; j < 1000; j++)

```

```

        {
            double x =
Globalus_kintamieji.posistemiu_isejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas] + j * mastelis;
            double y = Math.Exp(-(x -
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas]) * (x -
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas])) / (2 *
((Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas]) *
(Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas])) + Globalus_kintamieji.e));
            isejimo_narystes_funkciju_chart.Series[narystes_funkcijos_indeksas].Points.AddXY(x, y);
        }
        peles_nuspaudimo_vieta1_X = peles_nuspaudimo_vieta2_X;
    }
    else if (peles_nuspaudimo_vieta1_X > peles_nuspaudimo_vieta2_X)
    {
        skirtumas = peles_nuspaudimo_vieta1_X - peles_nuspaudimo_vieta2_X;

Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas] =
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas] - skirtumas * mastelis * 2;
        isejimo_narystes_funkciju_vidurkiu_textbox.Text =
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas].ToString();
        isejimo_narystes_funkciju_chart.Series[narystes_funkcijos_indeksas].Points.Clear();
        for (int j = 0; j < 1000; j++)
        {
            double x =
Globalus_kintamieji.posistemiu_isejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas] + j * mastelis;
            double y = Math.Exp(-(x -
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas]) * (x -
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas])) / (2 *
((Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas]) *
(Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas])) + Globalus_kintamieji.e));
            isejimo_narystes_funkciju_chart.Series[narystes_funkcijos_indeksas].Points.AddXY(x, y);
        }
        peles_nuspaudimo_vieta1_X = peles_nuspaudimo_vieta2_X;
    }
    if (peles_nuspaudimo_vieta1_Y < peles_nuspaudimo_vieta2_Y)
    {
        skirtumas = peles_nuspaudimo_vieta2_Y - peles_nuspaudimo_vieta1_Y;

Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas] =
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas] + skirtumas * mastelis * 2;
        isejimo_narystes_funkciju_neraiskumu_textbox.Text =
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas].ToString();
        isejimo_narystes_funkciju_chart.Series[narystes_funkcijos_indeksas].Points.Clear();
        for (int j = 0; j < 1000; j++)
        {
            double x =
Globalus_kintamieji.posistemiu_isejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas] + j * mastelis;
            double y = Math.Exp(-(x -
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas]) * (x -
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas])) / (2 *
((Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas]) *
(Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas])) + Globalus_kintamieji.e));
            isejimo_narystes_funkciju_chart.Series[narystes_funkcijos_indeksas].Points.AddXY(x, y);
        }
        peles_nuspaudimo_vieta1_Y = peles_nuspaudimo_vieta2_Y;
    }
    else if (peles_nuspaudimo_vieta1_Y > peles_nuspaudimo_vieta2_Y)
    {
        skirtumas = peles_nuspaudimo_vieta1_Y - peles_nuspaudimo_vieta2_Y;
    }
}

```



```

Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas] =
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas] - skirtumas * mastelis * 2;
isejimo_narystes_funkciju_neraiskumu_textbox.Text =
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas].ToString();
isejimo_narystes_funkciju_chart.Series[narystes_funkcijos_indeksas].Points.Clear();
for (int j = 0; j < 1000; j++)
{
    double x =
Globalus_kintamieji.posistemiu_isejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas] + j * mastelis;
    double y = Math.Exp(-(x -
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas]) * (x -
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas])) / (2 *
((Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas]) *
(Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
narystes_funkcijos_indeksas])) + Globalus_kintamieji.e));
    isejimo_narystes_funkciju_chart.Series[narystes_funkcijos_indeksas].Points.AddXY(x, y);
}
peles_nuspaudimo_vieta1_Y = peles_nuspaudimo_vieta2_Y;
}
}
}
private void isejimo_narystes_funkciju_chart_peles_atleidimo_ivykis(object sender, MouseEventArgs e)
{
    narystes_funkcijos_pasirinkimas = false;
}
//Taisyklės
private void taisykles_ir_checkbox_pazymejimo_pasikeitimo_ivykis(object sender, EventArgs e)
{
    taisykles_arba_checkbox.Checked = !taisykles_ir_checkbox.Checked;
}
private void taisykles_arba_checkbox_pazymejimo_pasikeitimo_ivykis(object sender, EventArgs e)
{
    taisykles_ir_checkbox.Checked = !taisykles_arba_checkbox.Checked;
}
private void taisykles_prideti_button_paspaudimo_ivykis(object sender, EventArgs e)
{
    int tikrinimas = 0;
    int tikrinimas2 = 0;
    for (int i = 0; i <=
Globalus_kintamieji.posistemiu_iejimu_skaicius[Globalus_kintamieji.mygtuko_tagas]; i++)
    {
        if (i < Globalus_kintamieji.posistemiu_iejimu_skaicius[Globalus_kintamieji.mygtuko_tagas])
        {
            if (taisykliu_listbox_masyvas[i].SelectedIndex != -1)
            {
                tikrinimas++;
            }
        }
        else
        {
            if (taisykliu_listbox_masyvas[i].SelectedIndex != -1)
            {
                tikrinimas2++;
            }
        }
    }
    if (tikrinimas>0&&tikrinimas2==1)
    {
        if (Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas.Max() > 0)
        {
            Funkcijos.masyvo_dydzio_keitimas_2d(ref
Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas,
Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length,
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas.Max()+1);
            Funkcijos.masyvo_dydzio_keitimas_3d(ref
Globalus_kintamieji.posistemiu_taisykliu_aktyviu_iejimu_indeksu_masyvas,
Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length,
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas.Max()+1,
Globalus_kintamieji.posistemiu_iejimu_skaicius.Max());
        }
    }
}

```

```

        Funkcijos.masyvo_dydzio_keitimas_3d(ref
Globalus_kintamieji.posistemiu_taisykliu_vidurkiu_masyvas,
Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length,
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas.Max()+1,
Globalus_kintamieji.posistemiu_iejimu_skaicius.Max());
        Funkcijos.masyvo_dydzio_keitimas_3d(ref
Globalus_kintamieji.posistemiu_taisykliu_neraiskumu_masyvas,
Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length,
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas.Max()+1,
Globalus_kintamieji.posistemiu_iejimu_skaicius.Max());
        Funkcijos.masyvo_dydzio_keitimas_2d(ref
Globalus_kintamieji.posistemiu_isejimu_taisykliu_vidurkiu_masyvas,
Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length,
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas.Max()+1);
        Funkcijos.masyvo_dydzio_keitimas_2d(ref
Globalus_kintamieji.posistemiu_isejimu_taisykliu_neraiskumu_masyvas,
Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length,
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas.Max()+1);
    }
    else
    {
        Funkcijos.masyvo_dydzio_keitimas_2d(ref
Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas,
Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length,
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas[Globalus_kintamieji.mygtuko_tagas] + 1);
        Funkcijos.masyvo_dydzio_keitimas_3d(ref
Globalus_kintamieji.posistemiu_taisykliu_aktyviu_iejimu_indeksu_masyvas,
Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length,
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas[Globalus_kintamieji.mygtuko_tagas] + 1,
Globalus_kintamieji.posistemiu_iejimu_skaicius[Globalus_kintamieji.mygtuko_tagas]);
        Funkcijos.masyvo_dydzio_keitimas_3d(ref
Globalus_kintamieji.posistemiu_taisykliu_vidurkiu_masyvas,
Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length,
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas[Globalus_kintamieji.mygtuko_tagas] + 1,
Globalus_kintamieji.posistemiu_iejimu_skaicius[Globalus_kintamieji.mygtuko_tagas]);
        Funkcijos.masyvo_dydzio_keitimas_3d(ref
Globalus_kintamieji.posistemiu_taisykliu_neraiskumu_masyvas,
Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length,
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas[Globalus_kintamieji.mygtuko_tagas] + 1,
Globalus_kintamieji.posistemiu_iejimu_skaicius[Globalus_kintamieji.mygtuko_tagas]);
        Funkcijos.masyvo_dydzio_keitimas_2d(ref
Globalus_kintamieji.posistemiu_isejimu_taisykliu_vidurkiu_masyvas,
Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length,
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas[Globalus_kintamieji.mygtuko_tagas] + 1);
        Funkcijos.masyvo_dydzio_keitimas_2d(ref
Globalus_kintamieji.posistemiu_isejimu_taisykliu_neraiskumu_masyvas,
Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length,
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas[Globalus_kintamieji.mygtuko_tagas] + 1);
    }
    Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas[Globalus_kintamieji.mygtuko_tagas]] = "Jei ";
    for (int i = 0; i <=
Globalus_kintamieji.posistemiu_iejimu_skaicius[Globalus_kintamieji.mygtuko_tagas]; i++)
    {
        if (i < Globalus_kintamieji.posistemiu_iejimu_skaicius[Globalus_kintamieji.mygtuko_tagas])
        {
            if (taisykliu_listbox_masyvas[i].SelectedIndex != -1)
            {
                Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas[Globalus_kintamieji.mygtuko_tagas]] =
Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas[Globalus_kintamieji.mygtuko_tagas]] +
Globalus_kintamieji.posistemiu_iejimu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas, i] + " " +
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas, i,
taisykliu_listbox_masyvas[i].SelectedIndex];

                Globalus_kintamieji.posistemiu_taisykliu_aktyviu_iejimu_indeksu_masyvas[Globalus_kintamieji.mygtuko_tagas,
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas[Globalus_kintamieji.mygtuko_tagas], i] = "on";

                Globalus_kintamieji.posistemiu_taisykliu_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas[Globalus_kintamieji.mygtuko_tagas], i] =
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas, i,
taisykliu_listbox_masyvas[i].SelectedIndex];

                Globalus_kintamieji.posistemiu_taisykliu_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas[Globalus_kintamieji.mygtuko_tagas], i] =

```

```

Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas, i,
taisykliu_listbox_masyvas[i].SelectedIndex];
    if (i <
Globalus_kintamieji.posistemiu_iejimu_skaicius[Globalus_kintamieji.mygtuko_tagas] - 1)
    {
        if (taisykliu_listbox_masyvas[i+1].SelectedIndex != -1)
        {
            if (taisykles_ir_checkbox.Checked)
            {
Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas[Globalus_kintamieji.mygtuko_tagas]] =
Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas[Globalus_kintamieji.mygtuko_tagas]] + " ir ";
            }
            else if (taisykles_arba_checkbox.Checked)
            {
Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas[Globalus_kintamieji.mygtuko_tagas]] =
Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas[Globalus_kintamieji.mygtuko_tagas]] + " arba ";
            }
        }
    }
}
else
{
Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas[Globalus_kintamieji.mygtuko_tagas]] =
Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas[Globalus_kintamieji.mygtuko_tagas]] + " tai " +
Globalus_kintamieji.posistemiu_isejimu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas] + " " +
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
taisykliu_listbox_masyvas[i].SelectedIndex];

Globalus_kintamieji.posistemiu_isejimu_taisykliu_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas[Globalus_kintamieji.mygtuko_tagas]] =
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas[Globalus_kintamieji.mygtuko_tagas,
taisykliu_listbox_masyvas[i].SelectedIndex];

Globalus_kintamieji.posistemiu_isejimu_taisykliu_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas[Globalus_kintamieji.mygtuko_tagas]] =
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraiskumu_masyvas[Globalus_kintamieji.mygtuko_tagas,
taisykliu_listbox_masyvas[i].SelectedIndex];
    }
    Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas[Globalus_kintamieji.mygtuko_tagas]++;
    taisykles_textbox.Clear();
    string Tekstas = "";
    for (int j = 0; j <
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas[Globalus_kintamieji.mygtuko_tagas]; j++)
    {
        Tekstas +=
Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas, j].ToString();
        Tekstas += Environment.NewLine;
    }
    taisykles_textbox.Text = Tekstas;
}
else
{
    MessageBox.Show("Netinkamas taisyklės parinkimas");
}
}
private void taisykles_isvalyti_button_paspaudimo_ivykis(object sender, EventArgs e)
{
    taisykles_textbox.Clear();
    for (int j = 0; j <
Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas[Globalus_kintamieji.mygtuko_tagas]; j++)
    {
        Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
j]=null;
        for(int i=0;i<
Globalus_kintamieji.posistemiu_iejimu_skaicius[Globalus_kintamieji.mygtuko_tagas]; i++)
        {

```

```

Globalus_kintamieji.posistemiu_taisykliu_aktyviu_iejimu_indeksu_masyvas[Globalus_kintamieji.mygtuko_tagas, j,i]
= null;
    }
    } Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas[Globalus_kintamieji.mygtuko_tagas]
= 0;
}
private void taisykles_isvalyti_pasirinkimus_button_paspaudimo_ivykis(object sender, EventArgs e)
{
    for (int i = 0; i <=
Globalus_kintamieji.posistemiu_iejimu_skaicius[Globalus_kintamieji.mygtuko_tagas]; i++)
    {
        taisykliu_listbox_masyvas[i].ClearSelected();
    }
}
//Meniu mygtukai
private void iejimai_button_Click(object sender, EventArgs e)
{
    if (Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
0] != null)
    {
        MessageBox.Show("Negalima keisti parametru tol, kol neišvalytos taisyklės");
        taisykles_button.PerformClick();
    }
    else
    {
        paslepti();
        iejimai_button.BackColor = Color.MediumSeaGreen;
        iejimu_sk_label.Show();
        iejimu_sk_textbox.Show();
        iejimu_pavadinimu_listbox_label.Show();
        iejimu_pavadinimu_listbox.Show();
        iejimu_pavadinimu_pervadinti_label.Show();
        iejimu_pavadinimu_pervadinti_textbox.Show();
        iejimu_diapozono_label.Show();
        iejimu_diapozono_min_label.Show();
        iejimu_diapozono_max_label.Show();
        iejimu_diapozono_min_textbox.Show();
        iejimu_diapozono_max_textbox.Show();
        iejimu_narystes_funkciju_skaiciaus_label.Show();
        iejimu_narystes_funkciju_skaiciaus_textbox.Show();
    }
}
private void narystes_funkcijos_button_Click(object sender, EventArgs e)
{
    if (iejimu_sk_textbox.Text ==
"Neįvestas" | Globalus_kintamieji.posistemiu_iejimu_skaicius[Globalus_kintamieji.mygtuko_tagas] <= 0)
    {
        MessageBox.Show("Įėjimų skaičius lygus 0");
        iejimai_button.PerformClick();
        iejimu_sk_textbox.Focus();
    }
    else if
(Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas, 0] != null)
    {
        MessageBox.Show("Negalima keisti parametru tol, kol neišvalytos taisyklės");
        taisykles_button.PerformClick();
    }
    else if (iejimu_pavadinimu_listbox_pasirinktas_indeksas== -1)
    {
        MessageBox.Show("Nepasirinktas įėjimas");
        iejimai_button.PerformClick();
        iejimu_pavadinimu_listbox.Focus();
    }
    else
    {
        paslepti();
        narystes_funkcijos_button.BackColor = Color.MediumSeaGreen;
        narystes_funkciju_listbox_label.Show();
        narystes_funkciju_listbox.Show();
        narystes_funkciju_pervadinti_label.Show();
        narystes_funkciju_pervadinti_textbox.Show();
        narystes_funkciju_vidurkiu_label.Show();
        narystes_funkciju_vidurkiu_textbox.Show();
        narystes_funkciju_neraiskumu_label.Show();
        narystes_funkciju_neraiskumu_textbox.Show();
        iejimu_narystes_funkciju_chart.Show();
    }
}

```

```

        //pridėjimas
        narystes_funkciju_listbox_label.Text =
Globalus_kintamieji.posistemiu_iejimu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas];
        narystes_funkciju_listbox.Items.Clear();
        for(int i = 0; i <
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas,
iejimu_pavadinimu_listbox_pasirinktas_indeksas]; i++)
        {
narystes_funkciju_listbox.Items.Add(Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_pavadinimu_masyvas[G
lobalus_kintamieji.mygtuko_tagas, iejimu_pavadinimu_listbox_pasirinktas_indeksas, i]);
        }
        iejimu_narystes_funkciju_chart.ChartAreas.Clear();
        var chartarea = new ChartArea();
        chartarea.Name = "ChartArea1";
        iejimu_narystes_funkciju_chart.ChartAreas.Add(chartarea);
        iejimu_narystes_funkciju_chart.Titles.Clear();
        var charttitle = new Title();
        charttitle.Name = "Title1";
        iejimu_narystes_funkciju_chart.Legends.Clear();
        var chartlegend = new Legend();
        iejimu_narystes_funkciju_chart.Legends.Add(chartlegend);
        iejimu_narystes_funkciju_chart.Titles.Add(charttitle);
        iejimu_narystes_funkciju_chart.ChartAreas["ChartArea1"].AxisX.Minimum =
Globalus_kintamieji.posistemiu_iejimu_diapozonu_min_masyvas[Globalus_kintamieji.mygtuko_tagas,iejimu_pavadinimu_
listbox_pasirinktas_indeksas];
        iejimu_narystes_funkciju_chart.ChartAreas["ChartArea1"].AxisX.Maximum =
Globalus_kintamieji.posistemiu_iejimu_diapozonu_max_masyvas[Globalus_kintamieji.mygtuko_tagas,iejimu_pavadinimu_
listbox_pasirinktas_indeksas];
        iejimu_narystes_funkciju_chart.ChartAreas["ChartArea1"].AxisY.Minimum = 0;
        iejimu_narystes_funkciju_chart.ChartAreas["ChartArea1"].AxisY.Maximum = 1;
        iejimu_narystes_funkciju_chart.Titles["Title1"].Text =
Globalus_kintamieji.posistemiu_iejimu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,iejimu_pavadinimu_lis
tbox_pasirinktas_indeksas];
        iejimu_narystes_funkciju_chart.atnaujinimo_ivykis();
    }
}
private void duomenys_button_Click(object sender, EventArgs e)
{
    paslepti();
    duomenys_button.BackColor = Color.MediumSeaGreen;
    duomenys_label.Show();
    duomenys_textbox.Show();
    duomenys_ieskoti_button.Show();
    duomenys_isvalyti_button.Show();
    //
    if
(Globalus_kintamieji.posistemiu_isoriniu_duomenu_tekstinis_masyvas[Globalus_kintamieji.mygtuko_tagas]!=null)
    {
        duomenys_textbox.Text =
Globalus_kintamieji.posistemiu_isoriniu_duomenu_tekstinis_masyvas[Globalus_kintamieji.mygtuko_tagas];
    }
}
private void isejimas_button_Click(object sender, EventArgs e)
{
    if (Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas,
0] != null)
    {
        MessageBox.Show("Negalima keisti parametru tol, kol neišvalytos taisyklės");
        taisykles_button.PerformClick();
    }
    else
    {
        paslepti();
        isejimas_button.BackColor = Color.MediumSeaGreen;
        isejimo_label.Show();
        isejimo_textbox.Show();
        isejimo_diapozono_label.Show();
        isejimo_diapozono_min_label.Show();
        isejimo_diapozono_max_label.Show();
        isejimo_diapozono_min_textbox.Show();
        isejimo_diapozono_max_textbox.Show();
        isejimo_narystes_funkciju_skaiciaus_label.Show();
        isejimo_narystes_funkciju_skaiciaus_textbox.Show();
        isejimo_pavadinimu_listbox_label.Show();
    }
}

```

```

isejimo_pavadinimu_listbox.Show();
isejimo_narystes_funkciju_vidurkiu_label.Show();
isejimo_narystes_funkciju_vidurkiu_textbox.Show();
isejimo_narystes_funkciju_neraskumu_label.Show();
isejimo_narystes_funkciju_neraskumu_textbox.Show();
isejimo_narystes_funkciju_pavadinimu_pervadinti_label.Show();
isejimo_narystes_funkciju_pavadinimu_pervadinti_textbox.Show();
isejimo_narystes_funkciju_chart.Show();
//
if
(Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas] > 0)
{
    isejimo_pavadinimu_listbox.Items.Clear();
    for (int i = 0; i <
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas]; i++)
    {
        isejimo_pavadinimu_listbox.Items.Add(Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_pavadinimu_masyvas
[Globalus_kintamieji.mygtuko_tagas, i]);
    }
    isejimo_narystes_funkciju_chart_atnaujinimo_ivykis();
}
}
private void taisykles_button_Click(object sender, EventArgs e)
{
    if (iejimu_sk_textbox.Text == "Neįvestas" ||
Globalus_kintamieji.posistemiu_iejimu_skaicius[Globalus_kintamieji.mygtuko_tagas] <= 0)
    {
        MessageBox.Show("Įėjimų skaičius lygus 0");
        iejimai_button.PerformClick();
        iejimu_sk_textbox.Focus();
    }
    else
    {
        paslepti();
        taisykles_button.BackColor = Color.MediumSeaGreen;
        taisykles_label.Show();
        taisykles_textbox.Show();
        taisykles_ir_checkbox.Show();
        taisykles_arba_checkbox.Show();
        taisykles_isvalyti_pasirinkimus_button.Show();
        taisykles_prideti_button.Show();
        taisykles_isvalyti_button.Show();
        taisykles_ir_checkbox.Checked = true;
        //
        Array.Resize(ref
taisykliu_listbox_masyvas, Globalus_kintamieji.posistemiu_iejimu_skaicius[Globalus_kintamieji.mygtuko_tagas]+1);
        Array.Resize(ref taisykliu_antrasciu_masyvas,
Globalus_kintamieji.posistemiu_iejimu_skaicius[Globalus_kintamieji.mygtuko_tagas] + 1);
        for (int i = 0; i <=
Globalus_kintamieji.posistemiu_iejimu_skaicius[Globalus_kintamieji.mygtuko_tagas]; i++)
        {
            if(i < Globalus_kintamieji.posistemiu_iejimu_skaicius[Globalus_kintamieji.mygtuko_tagas])
            {
                taisykliu_antrasciu_masyvas[i] = new Label();
                taisykliu_antrasciu_masyvas[i].AutoSize = true;
                taisykliu_antrasciu_masyvas[i].Text =
Globalus_kintamieji.posistemiu_iejimu_pavadinimu_masyvas[Globalus_kintamieji.mygtuko_tagas, i];
                taisykliu_antrasciu_masyvas[i].Location = new Point(atstumas_is_kaires + i * 100,
taisykles_ir_checkbox.Location.Y + taisykles_ir_checkbox.Height + atstumas_is_virsaus_tarp);
                this.Controls.Add(taisykliu_antrasciu_masyvas[i]);
                taisykliu_listbox_masyvas[i] = new ListBox();
                taisykliu_listbox_masyvas[i].Size = new Size(95,95);
                taisykliu_listbox_masyvas[i].Location = new
Point(taisykliu_antrasciu_masyvas[i].Location.X, taisykliu_antrasciu_masyvas[i].Location.Y+
taisykliu_antrasciu_masyvas[i].Height+ atstumas_is_virsaus_tarp);
                this.Controls.Add(taisykliu_listbox_masyvas[i]);
                for(int j = 0; j <
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_sk_masyvas[Globalus_kintamieji.mygtuko_tagas, i]; j++)
                {
                    taisykliu_listbox_masyvas[i].Items.Add(Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_pavadinimu_masyva
s[Globalus_kintamieji.mygtuko_tagas, i, j]);
                }
            }
        }
    }
}
else

```



```

        tekstas_rasymui.Write(Globalus_kintamieji.antrasciu_x_koordinaciju_masyvas[i].ToString()
+ " ");
    }
    tekstas_rasymui.WriteLine("|");
}
else
{
    tekstas_rasymui.Write("0");
    tekstas_rasymui.WriteLine("|");
    tekstas_rasymui.Write("null");
    tekstas_rasymui.WriteLine("|");
    tekstas_rasymui.Write("0");
    tekstas_rasymui.WriteLine("|");
}
//posistemės
if (mygtuku_masyvo_dydis > 0)
{
    tekstas_rasymui.Write(mygtuku_masyvo_dydis.ToString() + " ");
    tekstas_rasymui.WriteLine("|");
    for (int i = 0; i < Globalus_kintamieji.posistemiu_pavadinimu_masyvas.Length; i++)
    {
        tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_pavadinimu_masyvas[i].ToString() +
" ");
    }
    tekstas_rasymui.WriteLine("|");
    for (int i = 0; i < Globalus_kintamieji.posistemiu_x_koordinaciju_masyvas.Length; i++)
    {
        tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_x_koordinaciju_masyvas[i].ToString()
+ " ");
    }
    tekstas_rasymui.WriteLine("|");
    for (int i = 0; i < Globalus_kintamieji.posistemiu_y_koordinaciju_masyvas.Length; i++)
    {
        tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_y_koordinaciju_masyvas[i].ToString()
+ " ");
    }
    tekstas_rasymui.WriteLine("|");
}
else
{
    tekstas_rasymui.Write("0");
    tekstas_rasymui.WriteLine("|");
    tekstas_rasymui.Write("null");
    tekstas_rasymui.WriteLine("|");
    tekstas_rasymui.Write("0");
    tekstas_rasymui.WriteLine("|");
    tekstas_rasymui.Write("0");
    tekstas_rasymui.WriteLine("|");
}
//posistemės dialogas
//iejimai
//iejimu skaicius
for (int i = 0; i < Globalus_kintamieji.posistemiu_iejimu_skaicius.Length; i++)
{
    tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_iejimu_skaicius[i].ToString() + "
");
}
tekstas_rasymui.WriteLine("|");
//iejimu pavadinimai
for (int i = 0; i < Globalus_kintamieji.posistemiu_iejimu_pavadinimu_masyvas.GetLength(0);
i++)
{
    for (int j = 0; j <
Globalus_kintamieji.posistemiu_iejimu_pavadinimu_masyvas.GetLength(1); j++)
    {
        if(Globalus_kintamieji.posistemiu_iejimu_pavadinimu_masyvas[i, j] != null)
        {
            tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_iejimu_pavadinimu_masyvas[i, j].ToString() + " ");
        }
        else
        {
            tekstas_rasymui.Write("null"+" ");
        }
    }
}
tekstas_rasymui.WriteLine("|");

```



```

tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_iejimu_skaicius.Max().ToString());
tekstas_rasymui.WriteLine("|");
//min diapozonas
for (int i = 0; i < Globalus_kintamieji.posistemiu_iejimu_diapozonu_min_masyvas.GetLength(0);
i++)
    {
        for (int j = 0; j <
Globalus_kintamieji.posistemiu_iejimu_diapozonu_min_masyvas.GetLength(1); j++)
            {
                tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_iejimu_diapozonu_min_masyvas[i,
j].ToString() + " ");
            }
        tekstas_rasymui.WriteLine("|");
        //max diapozonas
        for (int i = 0; i < Globalus_kintamieji.posistemiu_iejimu_diapozonu_max_masyvas.GetLength(0);
i++)
            {
                for (int j = 0; j <
Globalus_kintamieji.posistemiu_iejimu_diapozonu_max_masyvas.GetLength(1); j++)
                    {
                        tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_iejimu_diapozonu_max_masyvas[i,
j].ToString() + " ");
                    }
                tekstas_rasymui.WriteLine("|");
                //iejimų narystės funkcijų sk
                for (int i = 0; i <
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_sk_masyvas.GetLength(0); i++)
                    {
                        for (int j = 0; j <
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_sk_masyvas.GetLength(1); j++)
                            {
                                tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_sk_masyvas[i, j].ToString() + "
");
                            }
                        tekstas_rasymui.WriteLine("|");
                        //iejimų narystės funkcijų pavadinimai
                        for (int i = 0; i <
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_pavadinimu_masyvas.GetLength(0); i++)
                            {
                                for (int j = 0; j <
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_pavadinimu_masyvas.GetLength(1); j++)
                                    {
                                        for (int k = 0; k <
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_pavadinimu_masyvas.GetLength(2); k++)
                                            {
                                                if (Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_pavadinimu_masyvas[i, j,
k] != null)
                                                    {
                                                        tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_pavadinimu_masyvas[i, j,
k].ToString() + " ");
                                                    }
                                                else
                                                    {
                                                        tekstas_rasymui.Write("null" + " ");
                                                    }
                                            }
                                        }
                                    }
                                tekstas_rasymui.WriteLine("|");
                                // Iėjimų narystės funkcijų max
                                tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_pavadinimu_masyvas.GetLength(2));
                                tekstas_rasymui.WriteLine("|");
                                //iejimų narystės funkcijų vidurkiai
                                for (int i = 0; i <
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas.GetLength(0); i++)
                                    {
                                        for (int j = 0; j <
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas.GetLength(1); j++)
                                            {
                                                for (int k = 0; k <
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas.GetLength(2); k++)

```

```

        {
tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas[i, j,
k].ToString() + " ");
        }
    }
}
tekstas_rasymui.WriteLine("|");
//iejimų narystės funkcijų neraiškumai
for (int i = 0; i <
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas.GetLength(0); i++)
{
    for (int j = 0; j <
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas.GetLength(1); j++)
    {
        for (int k = 0; k <
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas.GetLength(2); k++)
        {
tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas[i, j,
k].ToString() + " ");
        }
    }
}
tekstas_rasymui.WriteLine("|");
//išėjimas
for (int i = 0; i < Globalus_kintamieji.posistemiu_isejimu_pavadinimu_masyvas.Length; i++)
{
    if (Globalus_kintamieji.posistemiu_isejimu_pavadinimu_masyvas[i] != null)
    {
tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_isejimu_pavadinimu_masyvas[i].ToString() + " ");
    }
    else
    {
        tekstas_rasymui.Write("null" + " ");
    }
}
tekstas_rasymui.WriteLine("|");
for (int i = 0; i < Globalus_kintamieji.posistemiu_isejimu_diapozonu_min_masyvas.Length; i++)
{
tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_isejimu_diapozonu_min_masyvas[i].ToString() + " ");
}
tekstas_rasymui.WriteLine("|");
for (int i = 0; i < Globalus_kintamieji.posistemiu_isejimu_diapozonu_max_masyvas.Length; i++)
{
tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_isejimu_diapozonu_max_masyvas[i].ToString() + " ");
}
tekstas_rasymui.WriteLine("|");
for (int i = 0; i < Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_sk_masyvas.Length;
i++)
{
tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_sk_masyvas[i].ToString() + " ");
}
tekstas_rasymui.WriteLine("|");
//išėjimo narystės funkcijų pavadinimai
for (int i = 0; i <
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_pavadinimu_masyvas.GetLength(0); i++)
{
    for (int j = 0; j <
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_pavadinimu_masyvas.GetLength(1); j++)
    {
        if (Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_pavadinimu_masyvas[i, j] !=
null)
        {
tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_pavadinimu_masyvas[i,
j].ToString() + " ");
        }
        else
        {
            tekstas_rasymui.Write("null" + " ");
        }
    }
}
}

```

```

    }
    tekstas_rasymui.WriteLine("|");
    //išėjimo narystės funkcijų maksimumas

tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_pavadinimu_masyvas.GetLength(1));
tekstas_rasymui.WriteLine("|");
//išėjimo narystės funkcijų vidurkiaiai
for (int i = 0; i <
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas.GetLength(0); i++)
{
    for (int j = 0; j <
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas.GetLength(1); j++)
    {
        tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas[i, j].ToString()
+ " ");
    }
}
tekstas_rasymui.WriteLine("|");
//išėjimo narystės funkcijų neraiškumai
for (int i = 0; i <
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraiškumu_masyvas.GetLength(0); i++)
{
    for (int j = 0; j <
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraiškumu_masyvas.GetLength(1); j++)
    {
        tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraiškumu_masyvas[i,
j].ToString() + " ");
    }
}
tekstas_rasymui.WriteLine("|");
//Taisyklės
//posistemiu taisykliu skaiciaus masyvas
for (int i = 0; i < Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas.Length; i++)
{
    tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas[i].ToString() + " ");
}
tekstas_rasymui.WriteLine("|");
////taisykliu max

//tekstas_rasymui.Write((Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas.Max()).ToString()+" ");
//tekstas_rasymui.WriteLine("|");
//posistemiu taisykliu pavadinimu masyvas
for (int i = 0; i < Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas.GetLength(0);
i++)
{
    for (int j = 0; j <
Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas.GetLength(1); j++)
    {
        if (Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas[i, j] != null)
        {
            tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas[i,
j].ToString() + ",");
        }
        else
        {
            tekstas_rasymui.Write("null" + ",");
        }
    }
}
tekstas_rasymui.WriteLine("|");
//taisykliu max

tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas.GetLength(1).ToString() + "
");
    tekstas_rasymui.WriteLine("|");
    //posistemiu_taisykliu_aktyviu_iejimu_indeksu_masyvas
    for (int i = 0; i <
Globalus_kintamieji.posistemiu_taisykliu_aktyviu_iejimu_indeksu_masyvas.GetLength(0); i++)
    {
        for (int j = 0; j <
Globalus_kintamieji.posistemiu_taisykliu_aktyviu_iejimu_indeksu_masyvas.GetLength(1); j++)
        {
            for (int k = 0; k <
Globalus_kintamieji.posistemiu_taisykliu_aktyviu_iejimu_indeksu_masyvas.GetLength(2); k++)

```

```

        {
            if (Globalus_kintamieji.posistemiu_taisykliu_aktyviu_iejimu_indeksu_masyvas[i, j, k]
!= null)
            {
                tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_taisykliu_aktyviu_iejimu_indeksu_masyvas[i, j,
k].ToString() + " ");
            }
            else
            {
                tekstas_rasymui.Write("null" + " ");
            }
        }
    }
    tekstas_rasymui.WriteLine("|");
    //posistemiu_taisykliu_vidurkiu_masyvas
    for (int i = 0; i < Globalus_kintamieji.posistemiu_taisykliu_vidurkiu_masyvas.GetLength(0); i++)
    {
        for (int j = 0; j < Globalus_kintamieji.posistemiu_taisykliu_vidurkiu_masyvas.GetLength(1);
j++)
        {
            for (int k = 0; k <
Globalus_kintamieji.posistemiu_taisykliu_vidurkiu_masyvas.GetLength(2); k++)
            {
                tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_taisykliu_vidurkiu_masyvas[i, j, k].ToString() + " ");
            }
        }
        tekstas_rasymui.WriteLine("|");
        //posistemiu_taisykliu_neraiskumu_masyvas
        for (int i = 0; i < Globalus_kintamieji.posistemiu_taisykliu_neraiskumu_masyvas.GetLength(0);
i++)
        {
            for (int j = 0; j <
Globalus_kintamieji.posistemiu_taisykliu_neraiskumu_masyvas.GetLength(1); j++)
            {
                for (int k = 0; k <
Globalus_kintamieji.posistemiu_taisykliu_neraiskumu_masyvas.GetLength(2); k++)
                {
                    tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_taisykliu_neraiskumu_masyvas[i,
j, k].ToString() + " ");
                }
            }
        }
        tekstas_rasymui.WriteLine("|");
        //posistemiu_isejimu_taisykliu_vidurkiu_masyvas
        for (int i = 0; i <
Globalus_kintamieji.posistemiu_isejimu_taisykliu_vidurkiu_masyvas.GetLength(0); i++)
        {
            for (int j = 0; j <
Globalus_kintamieji.posistemiu_isejimu_taisykliu_vidurkiu_masyvas.GetLength(1); j++)
            {
                tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_isejimu_taisykliu_vidurkiu_masyvas[i, j].ToString() + " ");
            }
        }
        tekstas_rasymui.WriteLine("|");
        //posistemiu_isejimu_taisykliu_neraiskumu_masyvas
        for (int i = 0; i <
Globalus_kintamieji.posistemiu_isejimu_taisykliu_neraiskumu_masyvas.GetLength(0); i++)
        {
            for (int j = 0; j <
Globalus_kintamieji.posistemiu_isejimu_taisykliu_neraiskumu_masyvas.GetLength(1); j++)
            {
                tekstas_rasymui.Write(Globalus_kintamieji.posistemiu_isejimu_taisykliu_neraiskumu_masyvas[i, j].ToString() + "
");
            }
        }
        tekstas_rasymui.WriteLine("|");
        //linijos
        int skaicius = 0;
        foreach (var linija in linijos)
        {

```

```

        tekstas_rasymui.Write(((linija.Start.X / atstumas_tarp_posistemiu_X) /*-
objekto_atstumas_is_kaires - 100*/).ToString()+" ");
        tekstas_rasymui.Write(((linija.Start.Y / atstumas_tarp_posistemiu_Y) /*-
objekto_atstumas_is_kaires - 35*/).ToString() + " ");
        tekstas_rasymui.Write(((linija.End.X / atstumas_tarp_posistemiu_X) /*-
objekto_atstumas_is_kaires*/).ToString() + " ");
        tekstas_rasymui.Write(((linija.End.Y / atstumas_tarp_posistemiu_Y) /*-
objekto_atstumas_is_kaires - 35*/).ToString() + " ");
        skaicius++;
    }
    tekstas_rasymui.WriteLine("|");
    //liniju skaicius
    tekstas_rasymui.Write(skaicius.ToString());
    tekstas_rasymui.WriteLine("|");
    tekstas_rasymui.Close();
    tekstas.Close();
}
}

private void atverti_ToolStripMenuItem_Click(object sender, EventArgs e)
{
    int size = -1;
    var paieskos_dialogas = new OpenFileDialog();
    paieskos_dialogas.Title = "Atidaryti FuzzyThis sistema";
    paieskos_dialogas.Filter = "Text files (*.txt)|*.txt";
    DialogResult result = paieskos_dialogas.ShowDialog(); // Show the dialog.
    if (result == DialogResult.OK) // Test result.
    {
        string file = paieskos_dialogas.FileName;
        try
        {
            string text = File.ReadAllText(file);
            size = text.Length;

            if (paieskos_dialogas.FileName != "" /*&& mygtuku_masyvo_dydis == 0 &&
antrasciu_masyvo_dydis == 0*/)
            {
                System.IO.StreamReader reader = new System.IO.StreamReader(paieskos_dialogas.FileName);
                string Text = reader.ReadToEnd();
                reader.Close();
                string[] paragrafai = new string[100];
                paragrafai = Text.Split('|');
                //antraštės
                int antrasciu_sk = int.Parse((paragrafai[0].Trim()));
                string[] antrasciu_pav = paragrafai[1].Trim().Split(' ');
                string[] antrasciu_kord_x =(paragrafai[2].Trim().Split(' '));
                //antrasiu atkūrimas
                for(int i = 0; i < antrasciu_sk; i++)
                {
                    Point a = Cursor.Position;
                    Cursor.Position = this.PointToScreen(new Point(250 *
int.Parse(antrasciu_kord_x[i].Trim()), 0));
                    prideti_lygio_pavadinima_ToolStripMenuItem_Click(sender, e);
                    Cursor.Position = a;
                    antrasciu_masyvas[i].Text = antrasciu_pav[i];
                }
                //posistemės
                int posistemiu_sk = int.Parse((paragrafai[3].Trim()));
                string[] posistemiu_pav = paragrafai[4].Trim().Split(' ');
                string[] posistemiu_x_kord= paragrafai[5].Trim().Split(' ');
                string[] posistemiu_y_kord = paragrafai[6].Trim().Split(' ');
                //posistemės atkūrimas
                for (int i = 0; i < posistemiu_sk; i++)
                {
                    Point a = Cursor.Position;
                    Cursor.Position = this.PointToScreen(new Point(250 *
int.Parse(posistemiu_x_kord[i].Trim()), 0));
                    prideti_posistema_ToolStripMenuItem_Click(sender, e);
                    Cursor.Position = a;
                    mygtuku_masyvas[i].TextAlign = ContentAlignment.TopLeft;
                    mygtuku_masyvas[i].Text = posistemiu_pav[i];
                    mygtuku_masyvas[i].Location = new
Point(int.Parse(posistemiu_x_kord[i])*atstumas_tarp_posistemiu_X+objekto_atstumas_is_kaires,int.Parse(posistemiu
_y_kord[i])*atstumas_tarp_posistemiu_Y+objekto_atstumas_i_apacia);
                }
                //posistemės dialogas
                string[] posistemiu_in_sk = paragrafai[7].Trim().Split(' ');
                string[] posistemiu_in_pav = paragrafai[8].Trim().Split(' ');

```

```

int posistemiu_in_pav_max = int.Parse(paragrafai[9].Trim());
string[] posistemiu_in_d_min = paragrafai[10].Trim().Split(' ');
string[] posistemiu_in_d_max = paragrafai[11].Trim().Split(' ');
string[] posistemiu_in_nf_sk = paragrafai[12].Trim().Split(' ');
string[] posistemiu_in_nf_pav= paragrafai[13].Trim().Split(' ');
int posistemiu_in_nf_max= int.Parse(paragrafai[14].Trim());
string[] posistemiu_in_nf_vid= paragrafai[15].Trim().Split(' ');
string[] posistemiu_in_nf_ner= paragrafai[16].Trim().Split(' ');
//masyvų dydžio keitimas
Array.Resize(ref Globalus_kintamieji.posistemiu_iejimu_skaicius, posistemiu_sk);
Funkcijos.masyvo_dydzio_keitimas_2d(ref
Globalus_kintamieji.posistemiu_iejimu_pavadinimu_masyvas, posistemiu_sk, posistemiu_in_pav_max);
Funkcijos.masyvo_dydzio_keitimas_2d(ref
Globalus_kintamieji.posistemiu_iejimu_diapozonu_min_masyvas, posistemiu_sk, posistemiu_in_pav_max);
Funkcijos.masyvo_dydzio_keitimas_2d(ref
Globalus_kintamieji.posistemiu_iejimu_diapozonu_max_masyvas, posistemiu_sk, posistemiu_in_pav_max);
Funkcijos.masyvo_dydzio_keitimas_2d(ref
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_sk_masyvas, posistemiu_sk, posistemiu_in_pav_max);
Funkcijos.masyvo_dydzio_keitimas_3d(ref
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_pavadinimu_masyvas, posistemiu_sk,
posistemiu_in_pav_max, posistemiu_in_nf_max);
Funkcijos.masyvo_dydzio_keitimas_3d(ref
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas, posistemiu_sk, posistemiu_in_pav_max,
posistemiu_in_nf_max);
Funkcijos.masyvo_dydzio_keitimas_3d(ref
Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas, posistemiu_sk,
posistemiu_in_pav_max, posistemiu_in_nf_max);
//Iėjimo duomenų atkūrimas
int ind = 0;
int ind2 = 0;
for (int i = 0; i < posistemiu_sk; i++)
{
    Globalus_kintamieji.posistemiu_iejimu_skaicius[i] = int.Parse(posistemiu_in_sk[i]);
    for (int j = 0; j < posistemiu_in_pav_max; j++)
    {
        if (posistemiu_in_pav[ind] != "null")
        {
            Globalus_kintamieji.posistemiu_iejimu_pavadinimu_masyvas[i, j] =
posistemiu_in_pav[ind];
        }
        else
        {
            Globalus_kintamieji.posistemiu_iejimu_pavadinimu_masyvas[i, j] = null;
        }
        Globalus_kintamieji.posistemiu_iejimu_diapozonu_min_masyvas[i, j] =
double.Parse(posistemiu_in_d_min[ind]);
        Globalus_kintamieji.posistemiu_iejimu_diapozonu_max_masyvas[i, j] =
double.Parse(posistemiu_in_d_max[ind]);
        Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_sk_masyvas[i, j] =
int.Parse(posistemiu_in_nf_sk[ind]);
        ind++;
        for(int k = 0; k < posistemiu_in_nf_max; k++)
        {
            if (posistemiu_in_nf_pav[ind2] != "null")
            {
                Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_pavadinimu_masyvas[i, j, k] =
posistemiu_in_nf_pav[ind2];
            }
            else
            {
                Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_pavadinimu_masyvas[i, j, k] = null;
            }
            Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_vidurkiu_masyvas[i,
j, k] = double.Parse(posistemiu_in_nf_vid[ind2]);
        }
        Globalus_kintamieji.posistemiu_iejimu_narystes_funkciju_neraiskumu_masyvas[i, j, k] =
double.Parse(posistemiu_in_nf_ner[ind2]);
        ind2++;
    }
}
}
//Išėjimas
string[] isejimo_pav = paragrafai[17].Trim().Split(' ');
string[] isejimo_d_min = paragrafai[18].Trim().Split(' ');
string[] isejimo_d_max = paragrafai[19].Trim().Split(' ');

```

```

string[] isejimo_nf_sk = paragrafai[20].Trim().Split(' ');
string[] isejimo_nf_pav = paragrafai[21].Trim().Split(' ');
int isejimo_nf_sk_max = int.Parse(paragrafai[22].Trim());
string[] isejimo_nf_vid = paragrafai[23].Trim().Split(' ');
string[] isejimo_nf_ner = paragrafai[24].Trim().Split(' ');
//išėjimo masyvų dydžio keitimas
Array.Resize(ref Globalus_kintamieji.posistemiu_isejimu_pavadinimu_masyvas,
posistemiu_sk);
Array.Resize(ref Globalus_kintamieji.posistemiu_isejimu_diapozonu_min_masyvas,
posistemiu_sk);
Array.Resize(ref Globalus_kintamieji.posistemiu_isejimu_diapozonu_max_masyvas,
posistemiu_sk);
Array.Resize(ref Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_sk_masyvas,
posistemiu_sk);
Funkcijos.masyvo_dydzio_keitimas_2d(ref
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_pavadinimu_masyvas, posistemiu_sk, isejimo_nf_sk_max);
Funkcijos.masyvo_dydzio_keitimas_2d(ref
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas, posistemiu_sk, isejimo_nf_sk_max);
Funkcijos.masyvo_dydzio_keitimas_2d(ref
Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraiskumu_masyvas, posistemiu_sk, isejimo_nf_sk_max);
//išėjimo atkūtimas
int ind3 = 0;
for (int i = 0; i < posistemiu_sk; i++)
{
    if (isejimo_pav[i] != "null")
    {
        Globalus_kintamieji.posistemiu_isejimu_pavadinimu_masyvas[i] = isejimo_pav[i];
    }
    else
    {
        Globalus_kintamieji.posistemiu_isejimu_pavadinimu_masyvas[i] = null;
    }
    Globalus_kintamieji.posistemiu_isejimu_diapozonu_min_masyvas[i] =
double.Parse(isejimo_d_min[i].Trim());
    Globalus_kintamieji.posistemiu_isejimu_diapozonu_max_masyvas[i] =
double.Parse(isejimo_d_max[i].Trim());
    Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_sk_masyvas[i] =
int.Parse(isejimo_nf_sk[i].Trim());
    for(int j=0;j < isejimo_nf_sk_max; j++)
    {
        if (isejimo_nf_pav[ind3] != "null")
        {
            Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_pavadinimu_masyvas[i, j] = isejimo_nf_pav[ind3];
        }
        else
        {
            Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_pavadinimu_masyvas[i, j] = null;
        }
        Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_vidurkiu_masyvas[i, j]
= double.Parse(isejimo_nf_vid[ind3]);
        Globalus_kintamieji.posistemiu_isejimu_narystes_funkciju_neraiskumu_masyvas[i,
j] = double.Parse(isejimo_nf_ner[ind3]);
        ind3++;
    }
}
//taisyklės
string[] taisykliu_sk = paragrafai[25].Trim().Split(' ');
int taisykliu_sk_max = int.Parse(paragrafai[27].Trim());
string[] taisykliu_pav = paragrafai[26].Trim().Split(',');
string[] taisykliu_in_ain = paragrafai[28].Trim().Split(' ');
string[] taisykliu_in_vid = paragrafai[29].Trim().Split(' ');
string[] taisykliu_in_ner = paragrafai[30].Trim().Split(' ');
string[] taisykliu_is_vid = paragrafai[31].Trim().Split(' ');
string[] taisykliu_is_ner = paragrafai[32].Trim().Split(' ');
//taisykliu masyvu dydziu keitimas
Array.Resize(ref Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas,
posistemiu_sk);
Funkcijos.masyvo_dydzio_keitimas_2d(ref
Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas, posistemiu_sk, taisykliu_sk_max);
Funkcijos.masyvo_dydzio_keitimas_3d(ref
Globalus_kintamieji.posistemiu_taisykliu_aktyviu_iejimu_indeksu_masyvas, posistemiu_sk, taisykliu_sk_max,
posistemiu_in_pav_max);
Funkcijos.masyvo_dydzio_keitimas_3d(ref
Globalus_kintamieji.posistemiu_taisykliu_vidurkiu_masyvas, posistemiu_sk, taisykliu_sk_max,
posistemiu_in_pav_max);

```

```

        Funkcijos.masyvo_dydzio_keitimas_3d(ref
Globalus_kintamieji.posistemiu_taisykliu_neraiskumu_masyvas, posistemiu_sk, taisykliu_sk_max,
posistemiu_in_pav_max);
        Funkcijos.masyvo_dydzio_keitimas_2d(ref
Globalus_kintamieji.posistemiu_isejimu_taisykliu_vidurkiu_masyvas, posistemiu_sk, taisykliu_sk_max);
        Funkcijos.masyvo_dydzio_keitimas_2d(ref
Globalus_kintamieji.posistemiu_isejimu_taisykliu_neraiskumu_masyvas, posistemiu_sk, taisykliu_sk_max);
        //taisykliu atkurimas
        int ind4 = 0;
        int ind5 = 0;
        for (int i = 0; i < posistemiu_sk; i++)
        {
            Globalus_kintamieji.posistemiu_taisykliu_skaiciaus_masyvas[i] =
int.Parse(taisykliu_sk[i].Trim());
            for (int j = 0; j < taisykliu_sk_max; j++)
            {
                for (int k = 0; k < posistemiu_in_pav_max; k++)
                {
                    if (taisykliu_in_aii[ind5] != "null")
                    {
Globalus_kintamieji.posistemiu_taisykliu_aktyviu_iejimu_indeksu_masyvas[i, j, k] = taisykliu_in_aii[ind5];
                    }
                    else
                    {
Globalus_kintamieji.posistemiu_taisykliu_aktyviu_iejimu_indeksu_masyvas[i, j, k] = null;
                    }
                    Globalus_kintamieji.posistemiu_taisykliu_vidurkiu_masyvas[i, j, k] =
double.Parse(taisykliu_in_vid[ind5].Trim());
                    Globalus_kintamieji.posistemiu_taisykliu_neraiskumu_masyvas[i, j, k] =
double.Parse(taisykliu_in_ner[ind5].Trim());
                    ind5++;
                }
                if (taisykliu_pav[ind4] != "null")
                {
                    Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas[i, j] =
taisykliu_pav[ind4];
                }
                else
                {
                    Globalus_kintamieji.posistemiu_taisykliu_pavadinimu_masyvas[i, j] = null;
                }
                Globalus_kintamieji.posistemiu_isejimu_taisykliu_vidurkiu_masyvas[i, j] =
double.Parse(taisykliu_is_vid[ind4].Trim());
                Globalus_kintamieji.posistemiu_isejimu_taisykliu_neraiskumu_masyvas[i, j] =
double.Parse(taisykliu_is_ner[ind4].Trim());
                ind4++;
            }
        }
        //linijos
        string[] linijos = paragrafai[33].Trim().Split(' ');
        int liniju_sk = int.Parse(paragrafai[34].Trim());
        //liniju atkurimas
        int a1 = 0; int b2 = 1; int c3 = 2; int d4 = 3;
        for(int i=0;i< liniju_sk; i++)
        {
            sujungti = true;
            tarpinis_stabdis_SS = true;
            pt1_X = int.Parse(linijos[a1].Trim());
            pt1_Y = int.Parse(linijos[b2].Trim());
            pt2_X = int.Parse(linijos[c3].Trim());
            pt2_Y = int.Parse(linijos[d4].Trim());
            posistemės_paspaudimas(sender, e);
            posistemės_paspaudimas(sender, e);
            tarpinis_stabdis_SS = false;
            sujungti = false;
            a1 =a1+4;b2=b2+4;c3=c3+4;d4=d4+4;
        }
        //*****
    }
}
catch (IOException)
{
    MessageBox.Show("Failas nenuskaitytas, patikrinkite failą");
}
}
}

```