

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PRAKTINĖS INFORMATIKOS KATEDRA

Rita Laurikietytė

**Tempiamų daugiasluoksnių konstrukcinių elementų
projektavimo proceso modeliavimas**

Magistro darbas

Darbo vadovas

doc. A.Lenkevičius

Kaunas
2004

TURINYS

1. ĮVADAS	3
2. ANALITINĖ DALIS	5
2.1. PROBLEMINĖS DALIES BENDRAS APTARIMAS	5
2.2. SPRENDŽIAMOJO UŽDAVINIO AKTUALUMAS IR SPRENDIMO VARIANTO PARINKIMAS.....	5
2.2.1. <i>Daugiasluoksnio konstrukcinio elemento modelis</i>	6
2.2.2. <i>Daugiasluoksnio konstrukcinio elemento modelio algoritminė realizacija</i>	9
2.2.3. <i>Daugiasluoksnio konstrukcinio elemento parametrų skaičiavimo programos apžvalga</i>	10
2.3. DARBO TIKSLAS	12
3. PROJEKTINĖ DALIS	13
3.1. REIKALAVIMAI PROJEKTUOJAMAI PROGRAMAI	13
3.1.1. <i>Funkcionavimo reikalavimai</i>	13
3.1.2. <i>Reikalavimai vartotojo sąsajai</i>	13
3.1.3. <i>Programos eksploatavimo aplinka</i>	14
3.1.4. <i>Apribojimai</i>	14
3.2. DUOMENŲ STRUKTŪRA	14
3.3. PROJEKTUOJAMOS SISTEMOS ARCHITEKTŪRA	16
3.3.1. <i>Sistemos struktūrinė schema</i>	16
3.3.2. <i>Programos funkcinė schema</i>	17
3.4. PROGRAMINIŲ MODULIŲ SPECIFIKACIJA	19
3.4.1. <i>„DKE“ modulis</i>	19
3.4.2. <i>Teorinės ir praktinės medžiagos pateikimas</i>	20
3.5. TESTAVIMO MEDŽIAGA	20
3.5.1. <i>Programos testavimas</i>	20
3.5.2. <i>Programos palyginimas su skaičiavimais nesinaudojant programa</i>	25
4. VARTOTOJO DOKUMENTACIJA	27
4.1. PROGRAMOS FUNKCINIS APRAŠYMAS	27
4.2. SISTEMOS VADOVAS	27
4.2.1. <i>Programos lango dalis „Pradiniai duomenys“</i>	30
4.2.2. <i>Programos lango dalis „Skaičiavimo rezultatai“</i>	32
4.2.3. <i>Programos lango dalis „Skaičiavimo rezultatų grafinis atvaizdavimas“</i>	33
4.2.4. <i>Programos pranešimai</i>	34
4.3. PROGRAMOS INSTALIAVIMO DOKUMENTAS.....	35
4.3.1. <i>Reikalinga techninė įranga</i>	35
4.4. SISTEMOS ADMINISTRATORIAUS VADOVAS	36
5. PRODUKTO KOKYBĖS ĮVERTINIMAS	37
6. IŠVADOS	38
7. LITERATŪRA	39
8. THE MODELLING OF DESIGN PROCESS OF THE STRETCHED MULTILAYER STRUCTURAL ELEMENTS (SUMMARY)	40
9. TERMINŲ IR SANTRAUPŲ ŽODYNAS	41
10. PRIEDAI	42

1. ĮVADAS

Pirmieji šiuolaikinių kompozitų prototipai atsirado daugiau kaip prieš 5000 metų prieš Kristaus gimimą. Kaip rodo H. Hodges [1] ir Schartz M.M. [2] tyrimai, todėl į saulėje išdžiovintas molio plytas buvo dedami susmulkinti akmenys arba organinės kilmės medžiagos. Jau tada buvo pastebėta, kad tokios plytos mažiau susitraukia ir sutrūkinėja. Ir dabar siekiant sumažinti betono susitraukimą ir supleišėjimą dedami įvairūs metaliniai ar polimeriniai plaušeliai. Grynas molis arba betonas esti nestiprus tempimui, o priedai šioms medžiagoms gali suteikti naujų savybių. Tikriausiai pirmieji dirbtiniai kompozitai atsirado statybos reikmėms. Senovės puodžiai taip pat žinojo, kad dirbtiniu būdu galima reguliuoti keraminių dirbinių šiluminės izoliacijos vertę. Jie į molį įmaišydavo organinių medžiagų, kurios deginant gaminį sudegdavo ir sudarydavo uždaras tuštumas, iš dalies prisipildžiusias pelenų. Dėl tokių uždaru porų labai padidėdavo indų sienelių šiluminė varža ir vanduo ilgiau išlikdavo šiltas arba šaltas. Organinių priedų kiekiai reguliuodavo (aišku, be jokių skaičiavimų) šias šiluminės savybes. Įdomu pažymėti, kad pastaraisiais metais tokiu pat principu Lietuvoje pradėtos gaminti mažesnio šiluminio laidumo plytos.

Daugelis tyrinėtojų pateikia įrodymų, kad 3-ajame tūkstantmetyje prieš Kristaus gimimą Egipte ir Mesopotamijoje iš bitumine smala suklijuotų nendrių buvo gaminamos valtys, konstruktyviai analogiškos laivams, kurie Nilo deltoje naudojami dar ir dabar. Tai - dabartinio stiklo plastiko, gaminamo iš dirbtinių komponentų - stiklo pluošto ir polimerų, prototipas. Nendrės ir ypač jų pluoštas - palyginti stipri ir lengva korėta medžiaga, o bituminė derva, suklijuojanti ir impregnuojanti iš tokių pluoštų sudarytą atitinkamos formos gaminį, teikė naujai medžiagai ne tik papildomo stiprumo, bet ir nelaidumo vandeniui. Šios dvi komponentės atskirai visų šių savybių neturi.

Tačiau maždaug per visą naujosios eros pirmąjį tūkstantmetį nesukurta jokių naujų lengvesnių kompozicinių medžiagų. Tik 1955-1960 metais sukūrus boro ir jo junginių su kitais mineralais pluoštus, gauti nauji kompozitai - boro plastikai, kurių stiprumas 10-15 kartų didesnis, nei statybinio plieno, o patys jie 3-4 kartus lengvesni. Septintajame praeito amžiaus dešimtmetyje, išradus ir panaudojus labai didelio tamprumo modulio ir stiprumo anglies pluoštus, gauti dar geresnių savybių kompozitai. Medžiagos konstrukcinius privalumus rodo jos stiprumo ar tamprumo modulio santykis su savitu svoriu.

Ypatingas kompozitų kūrimo šuolis įvyko praeito amžiaus 4-ojo dešimtmečio pradžioje. Panaudojant fenolines ir polieterines dervas ir kitus pluoštus sukurti pirmieji itin stiprūs kompozitai - labai didelio stiprumo plastikai.

Atitinkamai parenkant komponentų savybes ir jų kiekius galima gauti norimų savybių kompozitą. Pavyzdžiui, turint kompoziciją su didesniu stiklo pluošto kiekiu, galima gauti medžiagą, kurios santykinis stiprumas tempiant yra daugiau kaip 60 kartų didesnis už metalo, naudojamo automobilių kėbulų gamybai. Ypač ženklus ekonominis efektas pasiektas naudojant kompozitines medžiagas lėktuvų gamyboje.

Naudojant daugiasluoksnių konstrukcinių elementų (DKE) projektavimo metodiką [4], buvo atlikti įtempimų būvio tyrimai, įvairių veiksnių įtaka daugiasluoksnių konstrukcinių elementų stiprumui ir standumui. Šie tyrimai buvo atlikti su konstrukciniais elementais iš šalto kietėjimo stiklo plastikų bei epoksidinė dervos.

Šiame darbe siūloma mokomoji programa skirta daugiasluoksnių konstrukcinių elementų projektavimui, o jos pagrindiniais vartotojais galėtų būti mechanikos technologijos katedros studentai, magistrantai ir dėstytojai, tačiau šis produktas gali būti taikomas ir daugiasluoksnių konstrukcinių elementų (DKE) mechaninių charakteristikų skaičiavimui daugelyje realių uždavinių.

Ši mokomoji programa įgalina pasirinkti DKE tipą, pastarojo mechaninius ir geometrinius parametrus. Jos vykdymo metu galima stebėti visų suskaičiuotų DKE mechaninių charakteristikų vertes ir pagrindinių priklausomybių grafikus, o esant reikalingumui realizuotas ir norimų rezultatų spausdinimas. Be to yra galimybė stebėti tekstiname lange paaiškinimus apie pagrindinius veiksmus atliekamus su programos kintamaisiais.

2. ANALITINĖ DALIS

2.1. Probleminės dalies bendras aptarimas

Pastaruoju metu atskirų medžiagų panaudojimas vis dažniau netenkina išaugusių reikalavimų, naujai projektuojamoms šiuolaikinėms, itin sudėtingoms mechaninėms konstrukcijoms. Jau vien todėl, kad dauguma šiuolaikinių gaminių konstruojami iš kompozicinių medžiagų, ir tik jų panaudojimo dėka galima gaminiui suteikti būtinas technines, eksploatacines bei kitas savybes. Iš principo, sumaniai parinktas kompozitas – tai nauja medžiaga, kurios savybės kokybiškai ir kiekybiškai skiriasi nuo kiekvieno jo komponento, o jei ir pati konstrukcija sudaryta iš elementų, kurie savo ruožtu esti kompozitais, tai atsiranda visa eilė papildomų galimybių iš esmės optimizuoti pastarosios darbą suteikti jai reikiamų specialių savybių. Kompozicinių medžiagų naudojimas įvairiuose konstrukcijose įgalino formuluoti ir efektyviai spręsti klasikinį optimizavimo uždavinį, kurio sprendinys didžiausio stiprumo ir mažiausio svorio ar kainos konstrukcija.

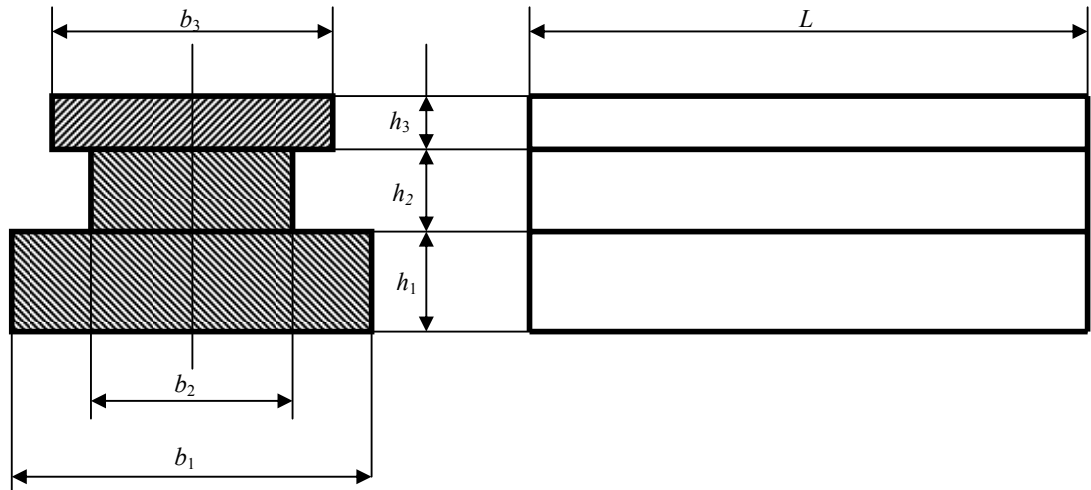
2.2. Sprendžiamojo uždavinio aktualumas ir sprendimo varianto parinkimas

Šiuo metu vis plačiau pramonėje taikomos šios kompozitinės medžiagos: stiklo plastikai, molio plastikai, bimetalai, grafito bei boro epoksidai ir kiti [3]. Svarbiausios šių kompozitų charakteristikos yra aukštas masės vieneto stiprumas. Kompozitai yra daugeliu atžvilgiu pranašesni už vienalytes medžiagas. Nors kiekvieną kompozitą sudarančios medžiagos turi įvairias (ir dažniausiai ne ypatingai aukštas) mechanines charakteristikas, tačiau gerai suprojektuotas kompozitas gali įgyti ypatingas charakteristikas, tenkinančias specifinius reikalavimus. Daugumoje atvejų, matematinis kompozito savybių modeliavimas, o algoritmizavus modelį - skaičiavimas yra sudėtingas uždavinys, tačiau modeliuojant kompozitą kaip daugiasluksnį strypą galima gauti nesudėtingą algoritmą pakankamai pilnai aprašantį pagrindines pastarojo funkcines (kaip konstrukcijos) savybes.

Modeliuojant konstrukciją sudarytą iš vienalyčių medžiagų kaip kompozitą, o kompozitą kaip daugiasluksnį strypą ir apsiribojant tik vienašio tempimo ir tiesinio tamprumo atvejais galima gauti paprastas tokios konstrukcijos parametrų matematinės išraiškas [4, 5] ir jas pateikti interaktyvios kompiuterinės programos pavidale. Būtent šiuo pagrįstas siūlomos taikomosios programos, pagrindinai skirtos studentų mokymui, aktualumas.

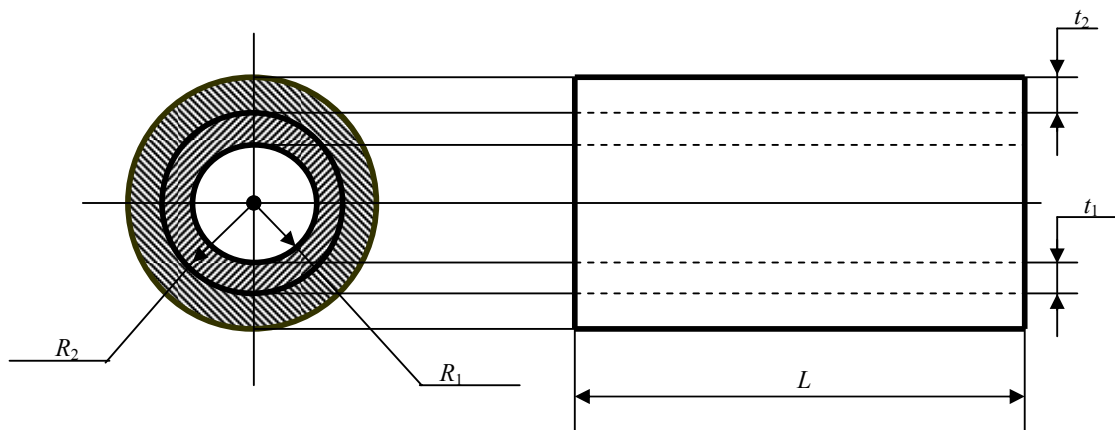
2.2.1. Daugiasluoksnio konstrukcinio elemento modelis

Tegul daugiasluoksnis konstrukcinis elementas (DKE) sudarytas iš n izotropinių sluoksnių (apribotų stačiais cilindriniais paviršiais), tempiamas išorinės apkrovos F kolinearios cilindrų sudaromosioms taip, kad jo galinės plokštumos lieka lygiagrečiomis. Esant tokioms prielaidoms visuose sluoksniuose esti centrinis tempimas, o DKE geometriją pilnai nusako jo skerspjūvio struktūra ir pastarojo ilgis. Žemiau nagrinėjamos trys skerspjūvio struktūros 1a, 1b ir 1c pav. (neužštrichuotos sritys reiškia cilindrinės tuštumas), o DKE skerspjūvis, tai



1a pav. Struktūra - sudaryta iš stačiakampių

dvimatė, nebūtinai susijusi sritis.

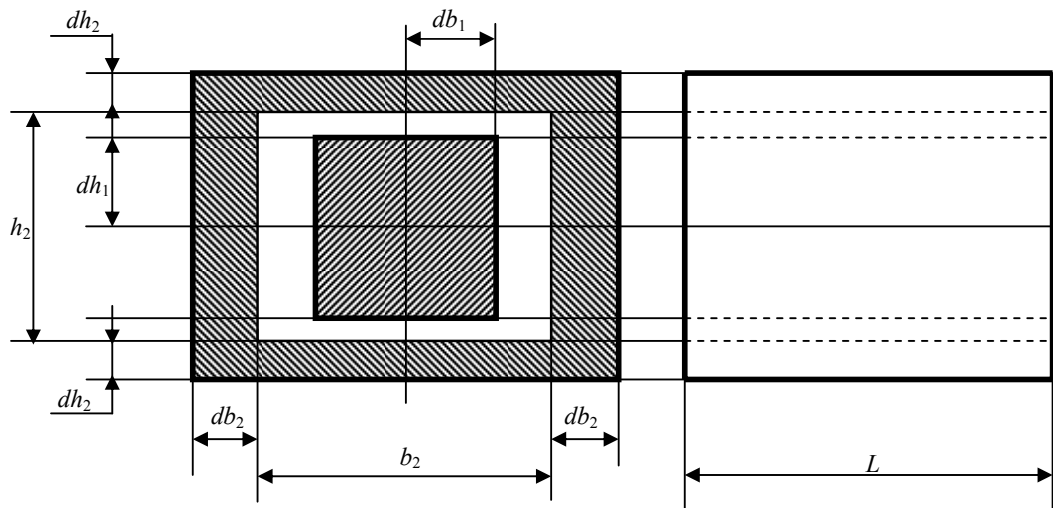


1b pav. Struktūra - žiedinė

Tokį DKE galima pilnai aprašyti (tiesinio tamprumo atveju) papildomai nusakant, visų sluoksnių skerspjūvio plotus A_i ir pastarųjų tamprumo modulius E_i

$$A_i, E_i, i = 1, 2, \dots, n, \quad (1)$$

čia n – sluoksnių skaičius konstrukciniame elemente.



1c pav. Struktūra – stačiakampė-žiedinė

Tegul visi DKE sluoksniai ir pats DKE visumoje tenkina Hooke'o dėsnį ir nepriklausomai nuo apkrovos dinaminio pobūdžio, pašalinus apkrovą, pilnai atstato savo pirminę formą bei tūrį, t.y. DKE ir jo sluoksnių $\sigma - \varepsilon$ diagramos – tiesės. Tokį kūną vadinsime tiesiškai tampriu. Atsižvelgiant į tai, kad atskiri DKE elementai negali deformuotis skirtingais dydžiais, jų absoliutūs pailgėjimai ir deformacijos lygios

$$\Delta L_i / L = \varepsilon_i \equiv \varepsilon, \quad i = 1, 2, \dots, n, \quad (2)$$

čia L - nedeformuoto DKE ilgis, o ΔL_i - pastarojo elementų absoliutūs pailgėjimai. Tegul DKE sudarytas iš n elementų tenkina Hooke'o dėsnį kaip vientisas kūnas, tada

$$\varepsilon E_K = N_K / A_K, \quad (3)$$

čia E_K ekvivalentusis DKE tamprumo modulis, N_K DKE kaip vientiso kūno įrašas, $\varepsilon \equiv \varepsilon_i, i = 1, 2, \dots, n$, deformacija, A_K viso DKE (išskyrus kiaurymes) skerspjūvio plotas

$$A_K = \sum_{i=1}^n A_i. \quad (4)$$

Jei N_i įrašos, atsirandančios atskiruose DKE elementuose, tada iš statinės pusiausvyros lygties turime

$$N_K = \sum_{i=1}^n N_i, \quad (5)$$

be to, kiekvienas DKE elementas tenkina Hooke'o dėsnį, tada

$$N_i = A_i E_i \varepsilon. \quad (6)$$

Pasinaudojus (3)-(6) galime išreikšti DKE ekvivalentųjį tamprumo modulį kaip elementų tamprumo modulių ir jų skerspjūvio plotų funkciją :

$$E_K = N_K / \varepsilon A_K = \sum_{i=1}^n A_i E_i / \sum_{i=1}^n A_i = \sum_{i=1}^n A_i E_i / A_K, \quad (7)$$

Pagal apibrėžimą DKE sudarančių elementų standumai

$$B_i = A_i E_i, \quad (8)$$

tada iš (7) turime, kad DKE standumas

$$B_K = A_K E_K = \sum_{i=1}^n B_i, \quad (9)$$

Iš (3), (8) ir (9) seka, kad DKE deformacija

$$\varepsilon = N_K E_K = F \sum_{i=1}^n A_i E_i, \quad (10)$$

o pasinaudojus (2) absoliutus DKE pailgėjimas

$$\Delta l = \varepsilon l = N_K l / B_K = Fl / \sum_{i=1}^n A_i E_i. \quad (11)$$

Išraiškos N_i turi tenkinti dvi sąlygas: statikos lygtį (pusiausvyros sąlyga)

$$\sum_{i=1}^n N_i = F \quad (12)$$

ir deformacijų darnos lygtį (vienodų visiems DKA elementams deformacijų sąlyga)

$$N_i / \sum_{i=1}^n A_i E_i = \varepsilon \quad arba \quad N_i = \varepsilon \sum_{i=1}^n A_i E_i, \quad (13)$$

nes pagal (2) ir (10), jie reiškia kiekvieno sluoksnio deformaciją ε . Jei duota apkrova F sistema (12)-(13) turi vienintelį sprendinį

$$N_i = F A_i E_i / \sum_{i=1}^n A_i E_i, \quad (14)$$

o, jei duota deformacija ε , sistema (12)-(13) turi vienintelį sprendinį

$$N_i = \varepsilon A_i. \quad (15)$$

Iš lygybės (9) randame įtempimus σ_i kiekviename DKE elemente

$$\sigma_i = F E_i / \sum_{i=1}^n A_i E_i. \quad (16)$$

2.2.2. Daugiasluoksnio konstrukcinio elemento modelio algoritminė realizacija

Jei DKE įtempto būvio vienintele priežastimi laikysime apkrovą F , o DKE sluoksnių skerspjūvių plotai A_i ir pastarųjų tamprumo moduliai E_i bei neapkrauto DKE ilgio l vertės yra žinomos, tai per pastarąsias galime išreikšti likusias:

1. Sluoksnių standumai $B_i = A_i E_i$,
2. DKE standumą $B_K = \sum_{i=1}^n B_i$,
3. Sluoksnių įrašas $N_i = F B_i / B_K$,
4. Sluoksnių normaliuosius įtempimus $\sigma_i = F E_i / B_K$,
5. DKE skerspjūvio plotą $A_K = \sum_{i=1}^n A_i$,
6. Ekvivalentųjį DKE tamprumo modulį $E_K = B_K / A_K$,
7. DKE (arba sluoksnio) deformaciją $\varepsilon = F / B_K$,
8. DKE absoliutų pailgėjimą $\Delta L = F L / B_K$,
9. Ekvivalentųjį DKE normalųjį įtempimą $\sigma_K = F / A_K$,
10. DKE įrašą F .

Jei DKE įtempto būvio vienintele priežastimi laikysime deformaciją ε , o DKE sluoksnių skerspjūvių plotai A_i ir pastarųjų tamprumo moduliai E_i bei nedeformuoto DKE ilgio L vertės yra žinomos, tai per pastarąsias galime išreikšti likusias:

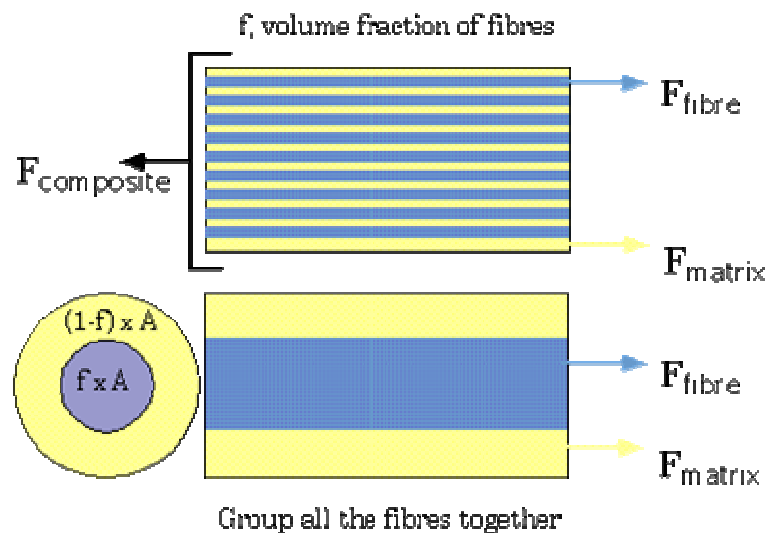
1. Sluoksnių standumai $B_i = A_i E_i$,

2. DKE standumą $B_K = \sum_{i=1}^n B_i$,
3. Sluoksnių įrašas $N_i = A_i E_i \varepsilon$,
4. Sluoksnių normaliuosius įtempimus $\sigma = E_i \varepsilon$,
5. DKE skerspjūvio plotą $A_K = \sum_{i=1}^n A_i$,
6. Ekvivalentų DKE tamprumo modulį $E_K = B_K / A_K$,
7. DKE (arba sluoksnio) deformaciją ε ,
8. DKE absoliutų pailgėjimą $\Delta L = L \varepsilon$,
9. Ekvivalentų DKE normalųjį įtempimą $E_K = B_K / A_K$,
10. DKE įrašą $F = B_K \varepsilon$.

2.2.3. Daugiasluoksnio konstrukcinio elemento parametrų skaičiavimo programos apžvalga

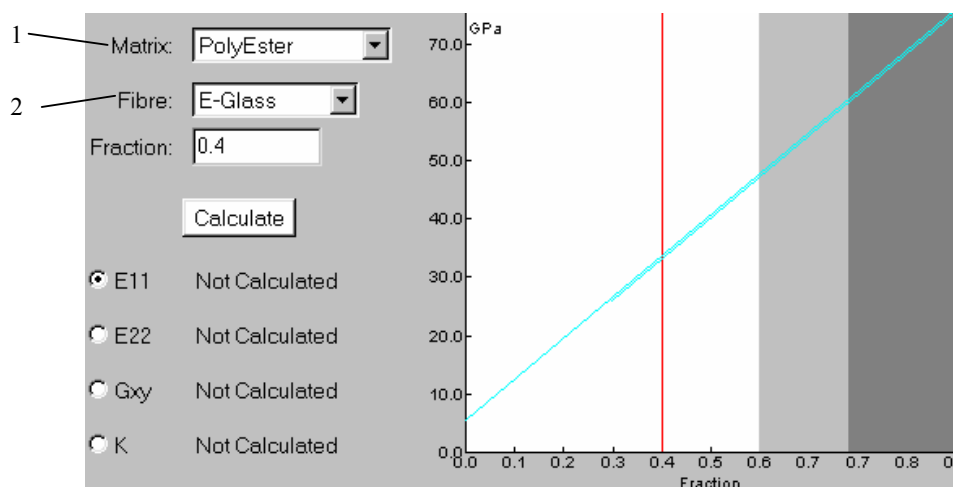
Trumpai aprašysime panašią programą. Ši programa sukurta Java kalboje ir yra laisvai pasiekama internete [6].

Programa analizuoja dviejų medžiagų kompozitą: pirmoji priskiriama matricai (Matrix), antroji – pluoštui (Fibre). Tiriamos konstrukcijos ir jos apkrovimo schema pateikta 2 pav. Pluošto ir audinio medžiagos dažnai turi labai skirtingus tamprumo modulius, todėl ir įtempimai jose esti skirtingi. Paprastai audinio tamprumo modulis parenkamas didesniu už matricos tamprumo modulį. Gerai žinomame kompozite – stiklo plastike, stiklo (pluošto) tamprumo modulis – 75 GPa, o poliesterio (matricos) – 5GPa.



2 pav. Konstrukcijos (kompozito) geometrijos ir apkrovos schema. $F_{composite}$ – tempimo jėga, A – viso kompozito skerspjūvio plotas, f – skerspjūvio ploto dalis, kurią užima pluoštas

Pagrindiniame programos lange (3 pav.) yra du išsiskleidžiantys sąrašų laukeliai, kuriuose galima pasirinkti: pirmąją medžiagą (Matrix), antrąją medžiagą (Fibre) ir laukelis antrosios medžiagos (Fibre) skerspjūvio ploto dalies visame kompozite įvedimui (Fraction).



3 pav. Pagrindinis programos langas:

1 - pirmosios medžiagos pasirinkimas; 2 - antrosios medžiagos pasirinkimas

Paspaudus mygtuką **Calculate** programa pagrindinio lango skaitinėje dalyje (jei tašku pažymėtas langelis **E11**) pateikia skaičiavimo rezultatus. Grafiniame lange rodoma suskaičiuotos kompozito tamprumo modulio vertės priklausomybė nuo antrosios (Fibre) skerspjūvio ploto dalies visame kompozite (Fraction). Vertikali (raudona) linija nurodo pasirinktą antrosios kompozito medžiagos skerspjūvio ploto dalį visame kompozite. Langelyje esančiame ties žyme **E11** matome skaitinę sukonstruoto kompozito tamprumo modulio vertę [GPa], t.y. mėlynos ir raudonos linijų susikirtimo taško ordinatę (3 pav.).

Aprašyta programa turi ir daugiau galimybių: apkrovos krypties keitimą pluošto orientacijos atžvilgiu, įtempimų tenzoriaus dedamųjų skaičiavimą ir kt.

Tuo tarpu siūloma programa turi platesnę konstrukcijos struktūrų pasirinkimo galimybę bei skaičiuoja ne tik konstrukcijos tamprumo modulį, bet ir daug kitų konstrukcijos bei jos elementų (sluoksnių) mechaninių parametrų, būtinų daugiasluoksnių konstrukcinių elementų projektavimui, todėl, manome, kad siūloma programa, jau bent apmokymo prasme, esti pranašesnė.

2.3. Darbo tikslas

Šio projekto tikslas - sukurti, naudojant šiuolaikines kompiuterines technologijas, lengvai įsisavinamą tempiamų daugiasluoksnių strypų mechaninių charakteristikų skaičiavimo programą, kuri įgalina studentą ar dėstytoją atsiriboti nuo rutininių skaičiavimų, koncentruojant dėmesį į pačių DKE projektavimą ir jų mechaninę esmę.

Siūloma programa, naudodama minimalų pradinių (įvedamų) duomenų skaičių, leidžia suskaičiuoti pagrindinius (būtinus DKE patikimai eksploatacijai) konstrukcijos mechaninius parametrus. Programai pakanka šių pradinių duomenų, aprašančių tik DKE sluoksnius (elementus): sluoksnių skerspjūvių plotų, sluoksnių medžiagų tamprumo modulių ir DKE kaip konstrukcijos: sluoksnių skaičiaus, DKE ilgio, DKE bendrosios apkrovos arba deformacijos vertės ir informacijos (loginė konstanta) nusakančios, ar duota apkrova veikianti konstrukciją ar pastarosios deformacija.

3. PROJEKTINĖ DALIS

3.1. Reikalavimai projektuojamai programai

3.1.1. Funkcionavimo reikalavimai

1. Programa turi veikti Windows 9x/ME/2000/XP operacinėse aplinkose.
2. Pilnam programos funkcionavimui būtina naršyklė Internet Explorer palaikanti HTML 4.0.
3. Programoje numatytos galimybės:
 - a) atlikti visus skaičiavimus numatytus analitinėje dalyje;
 - b) peržiūrėti skaičiavimo rezultatus;
 - c) gauti pranešimus apie duomenų įvedimo klaidas;
 - d) keisti pradinius duomenis;
 - e) esant reikalui, pakartoti skaičiavimus.
4. Programoje numatytos papildomos galimybės:
 - a) išsaugoti pradinius duomenis, skaičiavimo rezultatus, skaičiuotų funkcijų grafikus;
 - b) spausdinti skaičiavimo rezultatus, funkcijų grafikus ir diagramas;
 - c) turėti pagalbos meniu, trumpai aprašantį darbą su programa ir jos funkcijomis.

3.1.2. Reikalavimai vartotojo sąsajai

1. Vartotojo sąsaja (grafinio tipo) turi atvaizduoti įvestus pradinius duomenis.
2. Vartotojo sąsaja turi atvaizduoti skaičiavimo ir grafinius rezultatus gautus atlikus skaičiavimus.
3. Atsiradus duomenų įvedimo klaidoms, vartotojo sąsaja turi atvaizduoti pastarąsias specialiais pranešimas.
4. Programos valdymas turi būti paprastas ir nesunkiai įsisavinamas, todėl programoje naudojamos meniu eilutė ir priemonių juosta (menu tipo sąsaja), kurių pagalba vartotojas gali lengvai išsirinkti reikiamą komandą.
5. Programa skirta studentams, todėl pastaroji turi supažindinti vartotoją su jos valdymu, duomenų įvedimu, jų matavimo vienetais, skaičiuojamų daugiasluoksnių strypų geometrija (programos meniu juostoje išskirtas meniu punktas, kurį gali pasiskaityti vartotojas pirmą kartą dirbdamas su programa).

6. Programa turi būti patogi naudojimui (valdymas atliekamas pele ir/arba klavišų deriniais).
7. Programa turi būti suprantama vartotojui (visi tekstai ir paaiškinimai pateikti lietuvių kalba).
8. Vartotojui turi būti suteikta galimybė, bet kada baigti darbą (programa tai atlieka vartotojui sužadinus meniu punktą „Failai“ ir iš papildomo meniu išrinkus komandą „Baigti darbą su programa“ arba pasirinkus lango užvėrimo mygtuką).

3.1.3. Programos eksploatavimo aplinka

1. Vartotojo sąsaja pritaikyta dirbti labiausiai paplitusiose operacinėse aplinkose, t.y. Windows 9x, Me, 2000, XP.
2. Skaičiavimams sukurta originali taikomoji programa firmos „Microsoft“ objektinėje programavimo kalboje Visual Basic 6 (VB).
3. Praktinių užduočių ir pagrindinių teorinių žinių pasirinkimui naudojama standartinė naršyklė „Internet Explorer“, palaikanti HTML 4.0.

3.1.4. Apribojimai

1. Prieš įvedant duomenis ar atliekant skaičiavimus, operacinėje aplinkoje turi būti pasirinktas nacionalinis nustatymas - trupmeninės dešimtainio skaičiaus dalies skyrikliu pasirinktas „taškas“.

3.2. Duomenų struktūra

Pradinius duomenis, būtinus daugiasluoksnio strypo skaičiavimui tempimo atveju, galima suskirstyti į kelias grupes. Programoje naudojamos trys pradinių duomenų grupės: medžiagų mechaninės charakteristikos, skaičiuojamo daugiasluoksnio strypo geometriniai parametrai, apkrova arba deformacija.

1. Medžiagų mechaninės charakteristikos:
 - 1.1. Daugiasluoksnio strypo sluoksnių tamprumo moduliai (čia indeksas i reiškia sluoksnio numerį) E_i [MPa].
2. Skaičiuojamo bandinio geometrija:
 - 2.1. Strypo ilgis L [mm];

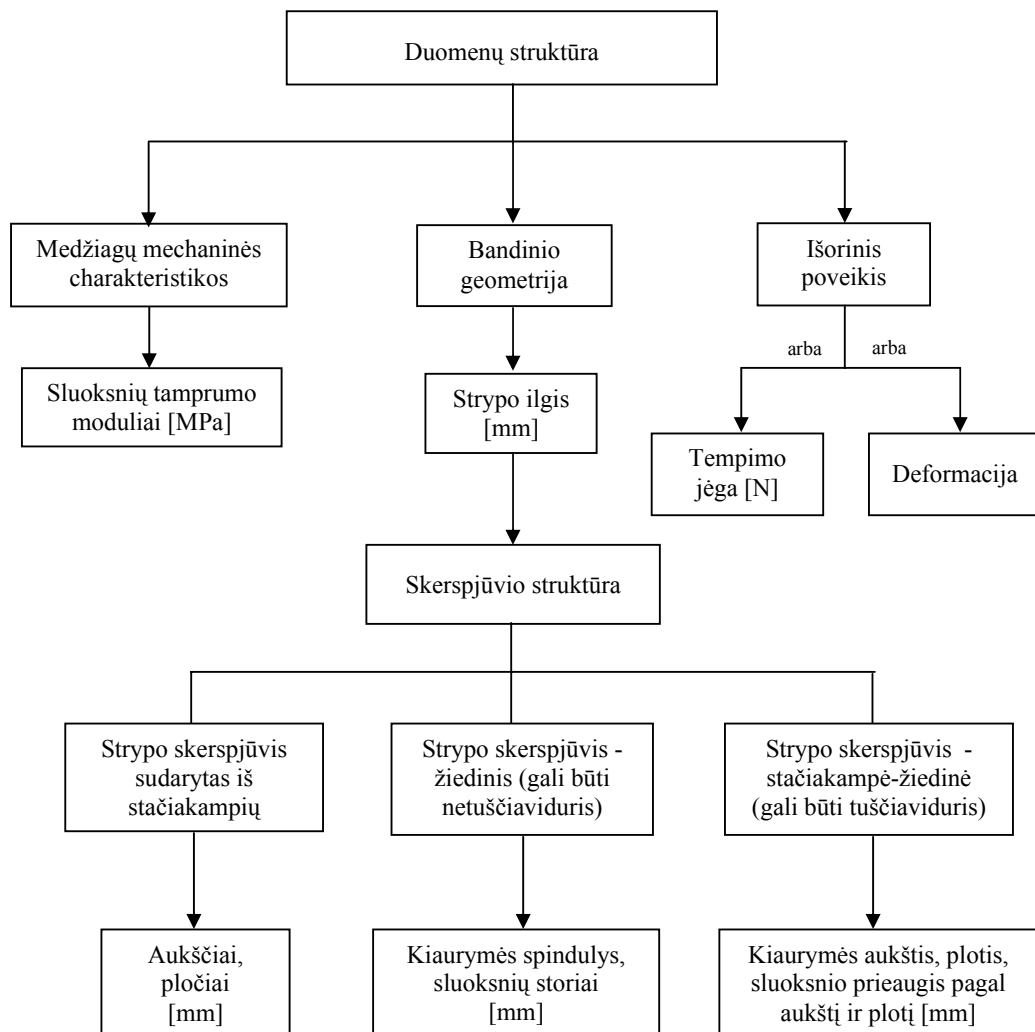
2.2. Skerspjuvio struktūra (atitinkanti skaičiuojamą daugiasluksnį strypą) ir jos parametrų vertės:

- sudaryta iš stačiakampių (*netaisyklinga*) - kiekvieno sluoksnio aukštis h_i [mm] ir plotis b_i [mm];
- žiedinė (*gali būti pilnavidurė*) - sluoksnio vidinis spindulys R_i [mm], jo storis t_i [mm];
- stačiakampė-žiedinė (*gali būti pilnavidurė*) - kiaurymės aukštis h_i [mm], plotis b_i [mm] ir sluoksnio storis pagal aukštį dh_i [mm] bei plotį db_i [mm].

3. Išorinis poveikis:

3.1. Tempimo jėga F [N] arba deformacija.

Programos pradinių duomenų struktūros grafinis vaizdas pateiktas 4 pav.

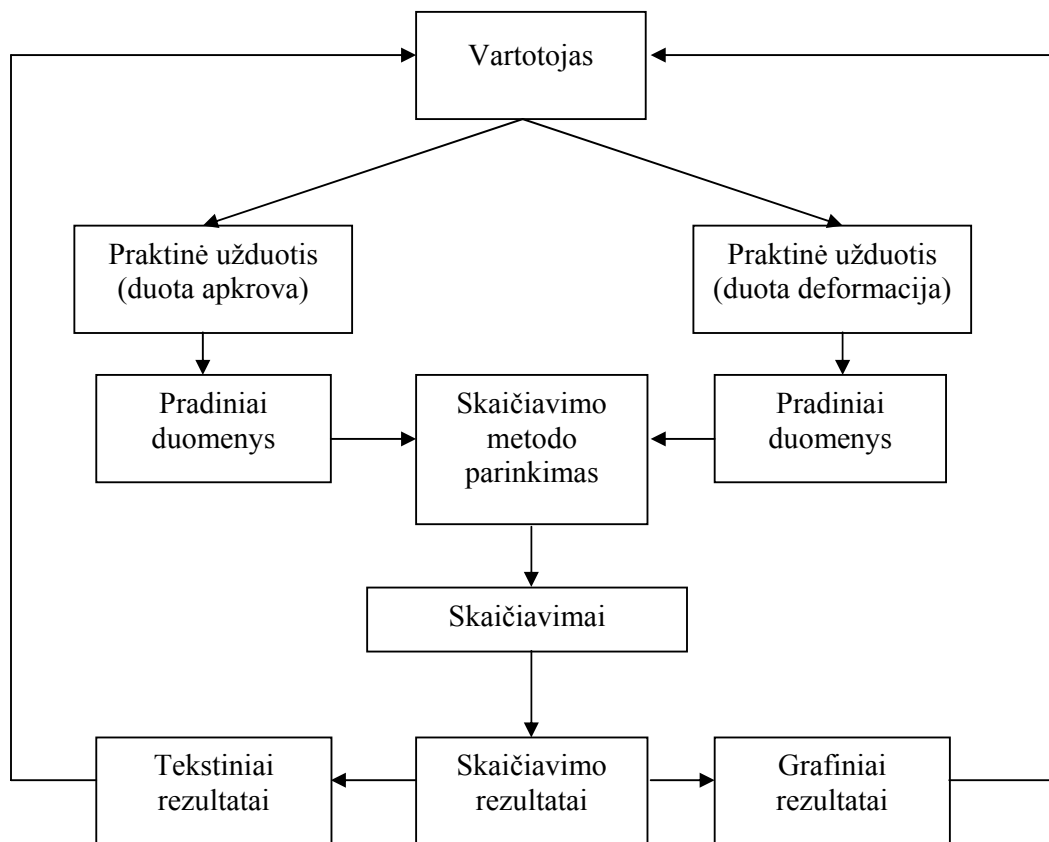


4 pav. Pradinių duomenų struktūra

3.3. Projektuojamos sistemos architektūra

3.3.1. Sistemos struktūrinė schema

Projektuojama sistema – tai daugiasluoksnis kompozitas, mūsų nagrinėjamu atveju – daugiasluoksnis strypas veikiamas ašinės apkrovos (tempimas – apkrova teigiama) arba ašinės deformacijos (tempimas - deformacija teigiama). Bendru atveju, tai sistema, kuri į poveikį (apkrovą arba deformaciją) duoda atsaką – deformaciją arba įrąžą. Tokios sistemos skaičiavimas – tai radimas jos įvairių parametru, kurie esti atsaku į minėtą poveikį. Pačią sistemą pilnai aprašo daugiasluoksnio strypo struktūra, sluoksnių skerspjūvių plotai, sluoksnių medžiagų tamprumo moduliai (medžiagų konstantos) ir sluoksnių ilgiai. Atsaką, atsiradus poveikiui (apkrovai arba deformacijai) sudaro šie kompozito parametrai: kompozito absoliutus pailgėjimas, kompozito deformacija (jei duota apkrova), kompozito įrąža, kompozito ekvivalentusis įtempimas bei šie sluoksnių parametrai: sluoksnių absoliutus pailgėjimas, sluoksnių deformacija (jei duota apkrova), sluoksnių įrąžos ir sluoksnių įtempimai.

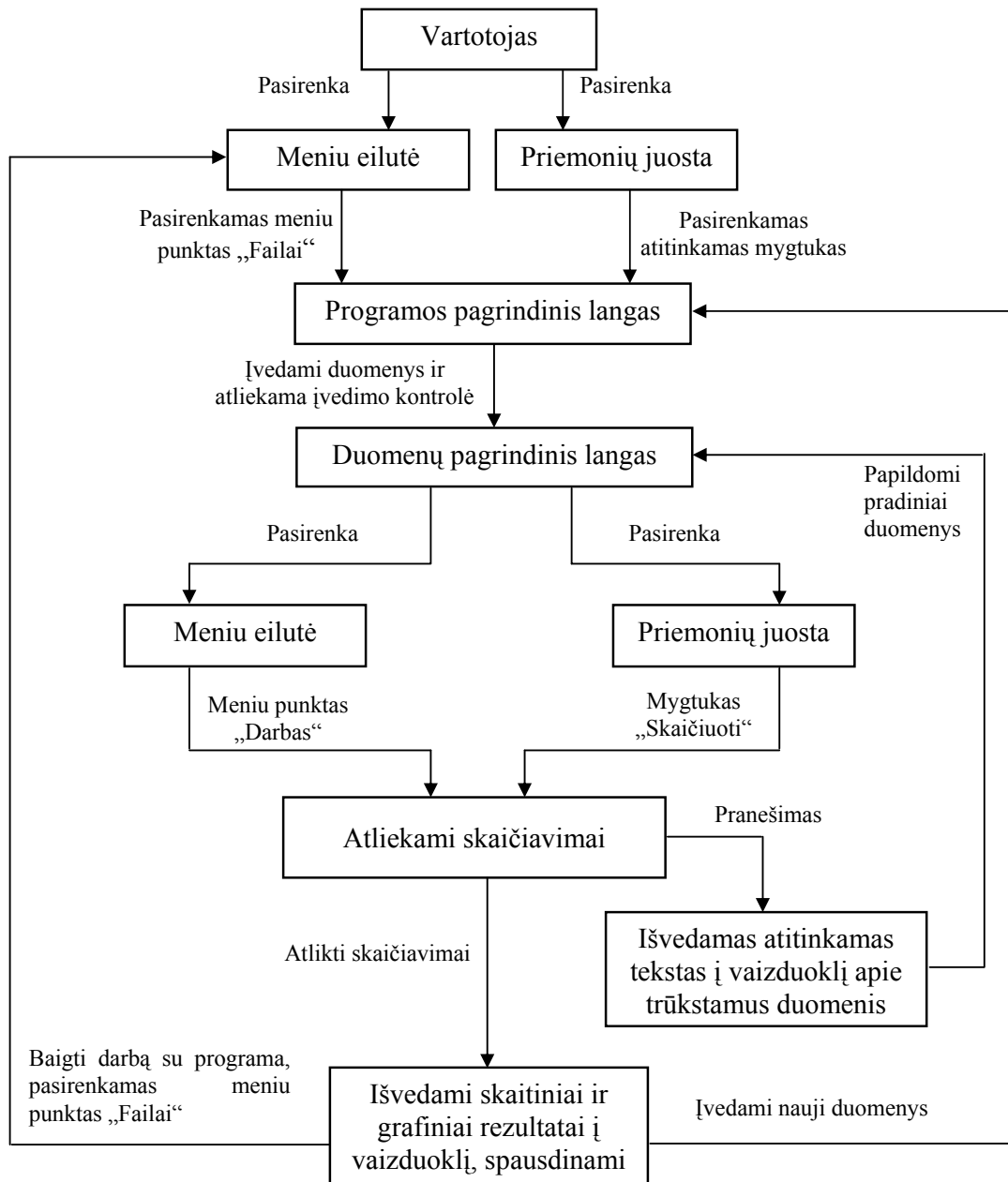


5 pav. Sistemos struktūrinė schema

5 pav. pateikta sistemos struktūrinė schema. Vartotojas gavęs praktinę užduotį faktiškai gauna tiriamos sistemos pilną aprašymą, įskaitant poveikio pobūdį (apkrovos vertę arba deformacijos vertę) ir informaciją apie tai, koku skaičiavimo metodu naudotis. Šie duomenys įvedami į programos, kuri ir atlieka išvardintų sistemos atsako parametrų verčių skaičiavimą ir jų perdavimą vartotojui, pradinių duomenų bloką.

3.3.2. Programos funkcinė schema

Programa atlieka tempiamų daugiasluoksnių strypų mechaninių parametrų verčių skaičiavimą. Programos funkcinė schema pateikta 6 pav. Vartotojas, naudodamas meniu



6 pav. Programos funkcinė schema

eilutę arba priemonių juostą gali atidaryti pagrindinį programos langą. Atsidariusiame lange esti galimybė įvesti užduoties pradinis duomenis, o juos įvedus programa atlieka pastarųjų kontrolę.

Užduoties vykdymui vartotojas gali pasirinkti vieną iš dviejų būdų:

1. Meniu punktas „Darbas“;
2. Priemonių juostoje esantis mygtukas „Skaičiuoti“.

Įvedus duomenis ir pasirinkus vieną minėtų būdų, atliekami skaičiavimai. Tuo atveju, jei vartotojas neįveda visų duomenų, gaunamas pranešimas apie padarytą klaidą. Tuomet, būtina grįžti į duomenų pagrindinį langą ir taisyti padarytą klaidą - papildyti įvedimo bloką trūkstamais duomenimis. Pataisius duomenis, pakartotinai atliekamas skaičiavimas vienu iš minėtų būdų.

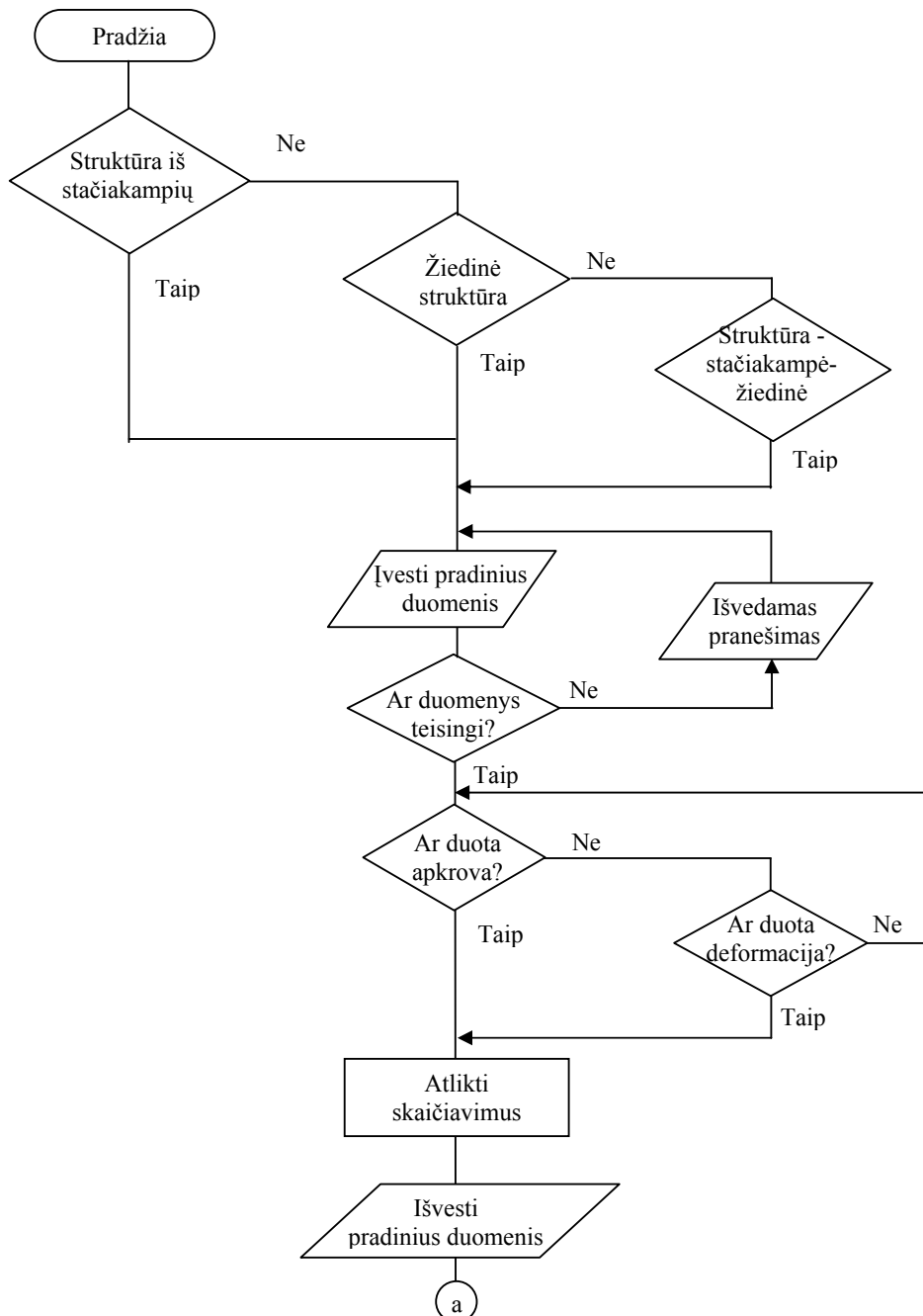
Jei pradiniai duomenys įvesti teisingai, tai atlikus skaičiavimą, programa vartotojui pateikia skaitinius bei grafinius rezultatus, kuriems numatyta spausdinimo galimybė.

Norint tęsti darbą, būtina grįžti į programos pagrindinį langą ir įvesti naujus pradinis duomenis. Priešingų atveju, pasirinkamas meniu punktas „Failai“ ir darbas baigiamas.

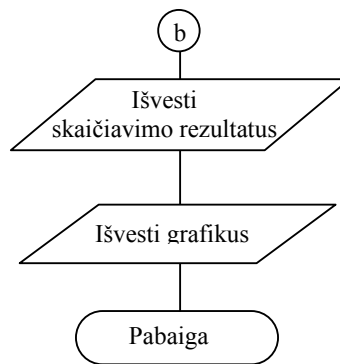
3.4. Programinių modulių specifikacija

3.4.1. „DKE“ modulis

Modulis DKE atlieka skaičiavimus pagal matematinės išraiškas, detalai aprašytas 2.2.2. skyriaus analitinėje dalyje, naudodamas užduoties pradinis duomenis. Be to, šis modulis atlieka ir įvestų pradinių duomenų kontrolę. Žemiau pateikiamas algoritmas, kurį realizuoja modulis DKE.



7 pav. „DKE“ modulio skaičiavimo algoritmas



7 pav. „DKE“ modulio skaičiavimo algoritmas

3.4.2. Teorinės ir praktinės medžiagos pateikimas

Teorinė medžiaga pateikiama nuosekliai, prisilaikant vadovėlio turinio. Taip pateikti teorinę medžiagą buvo nuspręsta konsultuojantis su dėstytojais, kurie skaito medžiagų mechanikos modulį. Skaitydamas teorinę medžiagą, vartotojas gali rasti pagrindinius apibrėžimus ir teorijos teiginius bei susijusias formules su praktine užduotimi.

Toks susistemintos teorijos pateikimas sutaupo vartotojui laiko, nes nereikia ieškoti papildomų šaltinių (knygų, konspektų), norint suprasti atliekamą praktinę užduotį.

Teorinę medžiagą ir praktines užduotis vartotojas jas gali rasti sužadinus meniu punktą „Pagalba“.

Praktinių užduočių variantai pateikti šio darbo 1 priede.

3.5. Testavimo medžiaga

3.5.1. Programos testavimas

Programos testavimui paruošti testinių pradinių duomenų ir atsakymų rinkiniai. Testinių rinkinių variantai:

- *duota apkrova;*
- *duota deformacija.*

Kontrolinis testas

1. Duota apkrova

1a lentelė. Pradiniai duomenys

Struktūra	E_i [MPa]	b_i [mm]	h_i [mm]	L_K [mm]	F_i [N]
Sudarytas iš stačiakampių	100	15	10	100	1000
	200	10	5		
	350	20	30		

1b lentelė. Skaičiavimo rezultatai

A_i [mm ²]	σ_{max} [MPa]	i_{max}	σ_i [MPa]	A_K [mm ²]	E_K [MPa]	ε_K	ΔL_K [mm]
150	1,489	3	0,426	800	293,750	0,00426	0,42553
50			0,851				
600			1,489				

Pradiniai duomenys 1a lentelė ir skaičiavimo rezultatai 1b lentelė pateikti programiniame lange (8 pav.).

Čia:

$$A[i] = A_i; e = \varepsilon; Sk = \sigma_K; Smax = \sigma_{max}; S[i] = \sigma_i; b[i] = b_i; dL = \Delta L_K, nmax = i_{max}.$$

SKAIČIAVIMO REZULTATAI

PRADINIAI DUOMENYS:

1-o sluoksnio plotis $b[1] = 15,000$ [mm]
 1-o sluoksnio aukštis $h[1] = 10,000$ [mm]
 1-o sluoksnio skerspjūvio plotas $A[1] = 150,000$ [mm²]
 1-o sluoksnio medžiagos tamprumo modulis $E[1] = 100,000$ [MPa]

2-o sluoksnio plotis $b[2] = 10,000$ [mm]
 2-o sluoksnio aukštis $h[2] = 5,000$ [mm]
 2-o sluoksnio skerspjūvio plotas $A[2] = 50,000$ [mm²]
 2-o sluoksnio medžiagos tamprumo modulis $E[2] = 200,000$ [MPa]

3-o sluoksnio plotis $b[3] = 20,000$ [mm]
 3-o sluoksnio aukštis $h[3] = 30,000$ [mm]
 3-o sluoksnio skerspjūvio plotas $A[3] = 600,000$ [mm²]
 3-o sluoksnio medžiagos tamprumo modulis $E[3] = 350,000$ [MPa]

Tempimo jėga $F = 1000,000$ [N]

SKAIČIAVIMO REZULTATAI:

Konstrukcijos skerspjūvio plotas $A_k = 800,000$ [mm²]
 Konstrukcijos tamprumo modulis $E_k = 293,750$ [MPa]

Normaliniai įtempimai S [MPa]:
 $S[1] = 0,426$
 $S[2] = 0,851$
 $S[3] = 1,489$

Santykinė deformacija $e = 0,00426$.
 Strypo pailgėjimas $dL = 0,42553$ [mm]

Maksimalus normalinis įtempimas $S_{max} = 1,489$ [MPa] yra 3 sluoksnyje.

8 pav. Pradiniai duomenys ir skaičiavimo rezultatai, kai duota apkrova ir struktūra sudaryta iš stačiakampių

2a lentelė. Pradiniai duomenys

Struktūra	E_i [MPa]	R_i , [mm]	t_i , [mm]	L_K [mm]	F , [N]
Žiedinė (pilnavidurė)	100	0	5	100	1000
	200	0	10		
	350	0	5		

2b lentelė. Skaičiavimo rezultatai

A_i [mm ²]	σ_{max} [MPa]	i_{max}	σ_i [MPa]	A_K [mm ²]	E_K [MPa]	ε_K	ΔL_K [mm]
78,540	1,074	3	0,307	1256,637	239,375	0,00307	0,30680
628,319			0,614				
549,779			1,074				

3a lentelė. Pradiniai duomenys

Struktūra	E_i [MPa]	b_i , [mm]	h_i , [mm]	db_i [mm]	dh_i [mm]	L_K [mm]	F , [N]
Stačiakampė- žiedinė	100	15	10	5	5	100	1000
	200	20	15	10	10		
	350	30	25	5	10		

3b lentelė. Skaičiavimo rezultatai

A_i [mm ²]	σ_{max} [MPa]	i_{max}	σ_i [MPa]	A_K [mm ²]	E_K [MPa]	ε_K	ΔL_K [mm]
300	0,824	3	0,235	1750	242,857	0,00235	0,23529
750			0,471				
700			0,824				

2. Duota deformacija

4a lentelė. Pradiniai duomenys

Struktūra	E_i [MPa]	b , [mm]	h , [mm]	L_K [mm]	ε
Sudarytas iš stačiakampių	100	15	10	100	0,05
	200	10	5		
	350	20	30		

4b lentelė. Skaičiavimo rezultatai

A_i [mm ²]	σ_{max} [MPa]	i_{max}	σ_i [MPa]	A_K [mm ²]	E_K [MPa]	F [N]	ΔL_K [mm]
150	17,500	3	5,000	800	293,75	1175,00	5,00000
50			10,000				
600			17,500				

Pradiniai duomenys 4a lentelė ir skaičiavimo rezultatai 4b lentelės pateikti programiniame lange (9 pav.).

Čia:

$$A[i] = A_i; e = \varepsilon; S_k = \sigma_K; S_{max} = \sigma_{max}; S[i] = \sigma_i; b[i] = b_i; dL = \Delta L_K, n_{max} = i_{max}.$$

SKAIČIAVIMO REZULTATAI

PRADINIAI DUOMENYS:

1-o sluoksnio plotis $b[1] = 15,000$ [mm]
 1-o sluoksnio aukštis $h[1] = 10,000$ [mm]
 1-o sluoksnio skerspjūvio plotas $A[1] = 150,000$ [mm²]
 1-o sluoksnio medžiagos tamprumo modulis $E[1] = 100,000$ [MPa]

2-o sluoksnio plotis $b[2] = 10,000$ [mm]
 2-o sluoksnio aukštis $h[2] = 5,000$ [mm]
 2-o sluoksnio skerspjūvio plotas $A[2] = 50,000$ [mm²]
 2-o sluoksnio medžiagos tamprumo modulis $E[2] = 200,000$ [MPa]

3-o sluoksnio plotis $b[3] = 20,000$ [mm]
 3-o sluoksnio aukštis $h[3] = 30,000$ [mm]
 3-o sluoksnio skerspjūvio plotas $A[3] = 600,000$ [mm²]
 3-o sluoksnio medžiagos tamprumo modulis $E[3] = 350,000$ [MPa]

Santykinė deformacija $e = 0,05000$

SKAIČIAVIMO REZULTATAI:

Konstrukcijos skerspjūvio plotas $A_k = 800,000$ [mm²]
 Konstrukcijos tamprumo modulis $E_k = 293,750$ [MPa]
 Tempimo jėga $F = 11750,000$ [N]

Normaliniai įtempimai S [MPa]:
 $S[1] = 5,000$
 $S[2] = 10,000$
 $S[3] = 17,500$

Strypo pailgėjimas $dL = 5,00000$ [mm]

Maksimalus normalinis įtempimas $S_{max} = 17,500$ [MPa] yra 3 sluoksnyje.

9 pav. Pradiniai duomenys ir skaičiavimo rezultatai, kai duota deformacija ir struktūra sudaryta iš stačiakampių

5a lentelė. Pradiniai duomenys

Struktūra	E_i [MPa]	R_i , [mm]	t_i , [mm]	L_K [mm]	F_s , [N]
Žiedinė	100	5	5	50	0,04
	200	10	15		

5b lentelė. Skaičiavimo rezultatai

A_i [mm ²]	σ_{max} [MPa]	i_{max}	σ_i [MPa]	A_K [mm ²]	E_K [MPa]	F [N]	ΔL_K [mm]
392,699	8,000	2	4,000	2513,274	187,375	18535,397	2,000
2120,575			8,000				

6a lentelė. Pradiniai duomenys

Struktūra	E_i [MPa]	b_i , [mm]	h_i , [mm]	db_i [mm]	dh_i [mm]	L_K [mm]	F , [N]
Stačiakampė- žiedinė	100	10	15	5	10	200	0,01
	200	20	35	10	20		

6b lentelė. Skaičiavimo rezultatai

A_i [mm ²]	σ_{max} [MPa]	i_{max}	σ_i [MPa]	A_K [mm ²]	E_K [MPa]	F [N]	ΔL_K [mm]
425,000	2,000	2	1,000	1575,000	173,016	2725,000	2,00000
1150,000			2,000				

Paiškinimai:

E_i – i -tojo sluoksnio medžiagos tamprumo modulis;

R_i – i -tojo sluoksnio vidinės kiaurymės spindulys;

b_i – i -tojo sluoksnio plotis;

h_i – i -tojo sluoksnio aukštis;

t_i – i -tojo sluoksnio storis;

F – DKE tempimo jėga;

db_i – i -tojo sluoksnio stačiakampio žiedo storis pagal plotį;

dh_i – i -tojo sluoksnio stačiakampio žiedo storis pagal aukštį;

A_K – konstrukcijos skerspjūvio plotas;

ε – DKE deformacija;

A_i – i -tojo sluoksnio skerspjūvio plotas;

E_K – DKE tamprumo modulis;

σ_{max} – didžiausias normalinis įtempimas sluoksnyje;

i_{max} – sluoksnio numeris, kuriame esti didžiausias įtempimas;

ΔL_K – DKE absoliutus pailgėjimas.

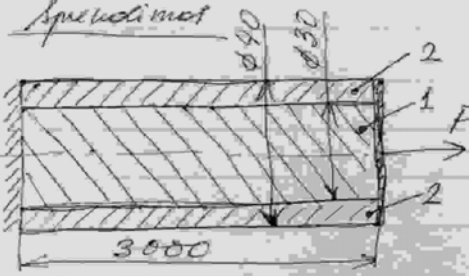
3.5.2. Programos palyginimas su skaičiavimais nesinaudojant programa

10a pav. pateikta tipinė daugiasluoksnio strypo skaičiavimo užduotis atlikta nesinaudojant programa.

Užduotis

Rasti cilindrinio tempiamo strypo, kurio išilginis pjūvis parodytas paveikėlyje, tempimas vidiniame slapyje 1 bei išoriniame slapyje 2, kai apkrova $F = 10^5 \text{ N}$, $d_1 = 30 \text{ mm}$, $d_2 = 40 \text{ mm}$, $E_1 = 1,2 \cdot 10^5 \text{ MPa}$ (varis) ir $E_2 = 2 \cdot 10^5 \text{ MPa}$ (plius). Nubraižuoti strypo eliovalutinį tempiamo modulį E_k ir strypo absoliutinį prailgėjimą, jei strypo ilgis $l = 3 \cdot 10^3 \text{ mm}$.

Sprendimas



Ausbraižinėme strypo šluoksnis šerėpjūvio plotus

$$A_1 = \frac{\pi d_1^2}{4} = 0,785 \cdot 900 = 706,856 \text{ mm}^2$$

$$A_2 = \frac{\pi (d_2^2 - d_1^2)}{4} = 0,785 \cdot (1600 - 900) = 549,479 \text{ mm}^2$$

ir viso strypo plotą

$$A_k = A_1 + A_2 = 706,856 + 549,479 = 1256,337 \text{ mm}^2$$

Strypo standumas

$$B_k = \sum_{i=1}^2 B_i = A_1 E_1 + A_2 E_2 = 706,856 \cdot 1,2 \cdot 10^5 + 549,479 \cdot 2 \cdot 10^5 = 84822720 + 109895742,87 = 194718462,87 \text{ Pa}$$

Tempimui vidiniame slapyje 1

$$\sigma_1 = \frac{F}{B_k} \cdot E_1 = \frac{10^5}{194718462,87} \cdot 1,2 \cdot 10^5 = 61,608 \text{ MPa}$$

Tempimui išoriniame slapyje 2

$$\sigma_2 = \frac{F}{B_k} \cdot E_2 = \frac{10^5}{194718462,87} \cdot 2 \cdot 10^5 = 102,681 \text{ MPa}$$

Eliaovalutinį viso strypo tempiamo modulį E_k

$$E_k = \frac{B_k}{A_k} = \frac{194718462,87}{1256,337} = 154999,783 \text{ MPa}$$

Absoliutinį deformacijai dydį $\Delta l = \frac{Fl}{B_k} = \frac{10^5 \cdot 3000}{194718462,87} = 1,54 \text{ mm}$

Atsakymas: $\sigma_1 = 61,608 \text{ MPa}$;
 $\sigma_2 = 102,681 \text{ MPa}$; $\Delta l = 1,54 \text{ mm}$

10a pav. Užduotis atlikta nesinaudojant programa

Ta pati užduotis atlikta naudojant programą.

SKAIČIAVIMO REZULTATAI

PRADINIAI DUOMENYS:

1-o sluoksnio vidinis spindulys $R_v[1] = 0,000$ [mm]
1-o sluoksnio išorinis spindulys $R_i[1] = 15,000$ [mm]
1-o sluoksnio skerspjūvio plotas $A[1] = 706,858$ [mm²]
1-o sluoksnio medžiagos tamprumo modulis $E[1] = 120000,000$ [MPa]

2-o sluoksnio vidinis spindulys $R_v[2] = 15,000$ [mm]
2-o sluoksnio išorinis spindulys $R_i[2] = 20,000$ [mm]
2-o sluoksnio skerspjūvio plotas $A[2] = 549,779$ [mm²]
2-o sluoksnio medžiagos tamprumo modulis $E[2] = 200000,000$ [MPa]

Tempimo jėga $F = 100000,000$ [N]

SKAIČIAVIMO REZULTATAI:

Konstrukcijos skerspjūvio plotas $A_k = 1256,637$ [mm²]
Konstrukcijos tamprumo modulis $E_k = 155000,000$ [MPa]

Normaliniai įtempimai S [MPa]:
 $S[1] = 61,608$
 $S[2] = 102,681$

Santykinė deformacija $e = 0,00051$.
Strypo pailgėjimas $dL = 0,15402$ [mm]

Maksimalus normalinis įtempimas $S_{max} = 102,681$ [MPa] yra 2 sluoksnyje.

10b pav. Užduotis atlikta naudojant programą

4. VARTOTOJO DOKUMENTACIJA

4.1. Programos funkcinis aprašymas

Daugiasluoksnių konstrukcijų elementų modeliavimo programa skirta KTU Panevėžio instituto Technologijos fakulteto studentams. Su šia programa galima atlikti studentams skirtas praktines užduotis bei spręsti kai kuriuos praktikoje pasitaikančius uždavinius.

Naudojantis šia programa, paprastėja daugelis daugiasluoksnių strypų mechaninių charakteristikų skaičiavimo uždavinių. Tai leidžia studentui ar dėstytojui atsiriboti nuo rutininių skaičiavimų ir įsigilinti į DKE projektavimą bei jo mechaninę esmę.

Atsižvelgiant į užduoties sąlygą, vartotojas gali įvesti pradinius užduoties duomenis: pasirinkti tempiamo strypo struktūrą, sluoksnių skerspjūvių parametrus, sluoksnių medžiagas (jų tamprumo modulius), apkrovos arba deformacijos vertę bei strypo ilgį. Papildant įvedimą nauju sluoksniu, „Pradinių duomenų lange“ atsiranda grafinis konstruojamo strypo vaizdas.

Programa sudaryta taip, kad vartotojas tame pačiame programos lange gali patikrinti suvestus duomenis, išanalizuoti gautus rezultatus, peržiūrėti pagal skaičiavimo rezultatus nubraižytus grafikus ir juos atspausdinti. Programoje numatyta galimybė išsaugoti pradinius duomenis, skaitinius ir grafinius rezultatus kompiuterio laikmenoje.

4.2. Sistemos vadovas

Kuriant šią programą buvo siekta sukurti, patogią, lengvai suprantamą vartotojui, neperkrautą ir lengvai valdomą programą. Todėl, vartotojas per trumpą laiką gali išmokti naudotis programa, kuri be to aprūpinta trumpa ir aiškia pagalba.

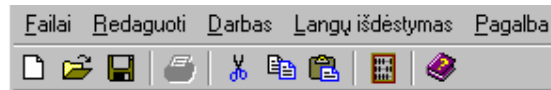
Vartotojo darbas pradedamas programos paleidimu (programa paleidžiama vykdomuoju failu „DKE.exe“). Startavus programai pasirodo pasirinkimų langas (11 pav). Pasirinkus vieną



11 pav Pasirinkimų langas

iš mygtukų galima: pradėti darbą, baigti darbą arba gauti trumpą informaciją apie programą.

Pasirinkus mygtuką „Pradėti darbą“ atsidaro pagrindinis langas, kuriame galima pasirinkti ar įkelti anksčiau sprestą užduotį, ar spresti naują. Tai galima padaryti dviem būdais (12 pav.): meniu eilutėje arba priemonių juostoje.



12 pav. Meniu eilutė ir priemonių juosta

Naudojimasis programa:

Naudodamas šią programą vartotojas veiksmų pasirinkimą gali atlikti meniu arba mygtukų pagalba. Meniu valdymas atliekamas pelės ir/arba klaviatūros pagalba:

- Norimą meniu punktą galima išsirinkti naudojantis pele:
 - dėl to reikia nuvesti pelės rodyklę ant norimo meniu punkto ir spausti kairįjį pelės klavišą ir iš papildomo meniu pele reikia išsirinkti norimą komandą. Po šių veiksmų programa atliks pasirinkto meniu punkto veiksmą;
- Norimą meniu punktą galima išsirinkti naudojantis klaviatūra:
 - dėl to reikia spausti klavišą *Alt* ir klavišą, atitinkantį pabrauktąjį meniu punkto raidę (pavyzdžiui *Alt+F* komandai „Failai“ sužadinti) ir iš papildomo meniu norimą komandą reikia išsirinkti naudojantis klaviatūros klavišais pažymėtais rodyklėmis ↑, ↓. Judant papildomu meniu išsirinkti komandą (tampa nuspalvinta mėlyna spalva), o komandos vykdymas įvyksta spaudžiant klavišą „Enter“.
- Meniu eilutė
 - Failai** – meniu punkte galima išsaugoti, atidaryti, atspausdinti skaitinius ir grafinius rezultatus bei baigti darbą su programa:
 - **Naujas** (*Ctrl+N*) – spresti naują užduotį;
 - **Atidaryti** (*Ctrl+O*) – įkelti anksčiau skaičiuotą užduotį;
 - **Uždaryti** – uždaryti išspręstą užduotį;
 - **Saugoti...** - saugoti išspręstą užduotį faile;
 - **Spausdinti** - spausdinti skaitinius ir grafinius rezultatus;
 - **Baigti darbą su programa** – išeiti iš programos.
 - Redaguoti** – leidžia vartotojui iškirpti, kopijuoti, įstatyti pažymėtas dalis:
 - **Iškirpti** – iškirpti pažymėtą dalį;
 - **Kopijuoti** - kopijuoti pažymėtą dalį;

- **Įstatyti** – įdėti nukopijuotą dalį.

Darbas – leidžia vartotojui atlikti skaičiavimus:

- **Skaičiuoti** – atlikti skaičiavimus.

Išdėstymas langų - leidžia vartotojui pasirinkti langų išdėstymo tvarką:










- **Kaskada** – langų išdėstymas pakopomis;
- **Vertikaliai** – vertikalus langų išdėstymas;
- **Horizontaliai** – horizontalus langų išdėstymas.

Pagalba - vartotojui skirta pagalba:

- **Turinys** – suteikia informaciją apie programos meniu eilutę bei priemonių juostą ir apie pradinių duomenų įvedimą, skaičiavimo rezultatus, jų grafinį vaizdavimą. Turinyje pateikta ir teorinė medžiaga bei praktinės užduotys.

Žymiai greičiau norimą komandą galima įvykdyti, pasinaudojus priemonių juosta, kurioje sukurti dažniausiai naudojamų komandų mygtukai. Norint sužinoti kokį veiksmą atlieka kiekvienas priemonių juostoje esantis mygtukas, tereikia priartinti pelės rodyklę prie mygtuko - gaunamas tekstinis pranešimas apie mygtuko atliekamus veiksmus.

▪ Priemonių juosta

-  - spręsti naują užduotį;
-  - įkelti anksčiau skaičiuotą užduotį;
-  - saugoti užduotį faile;
-  - spausdinti rezultatus;
-  - iškirpti pažymėtą dalį;
-  - kopijuoti pažymėtą dalį;
-  - įdėti nukopijuotą dalį;
-  - atlikti skaičiavimą;
-  - iškviešti pagalbą.

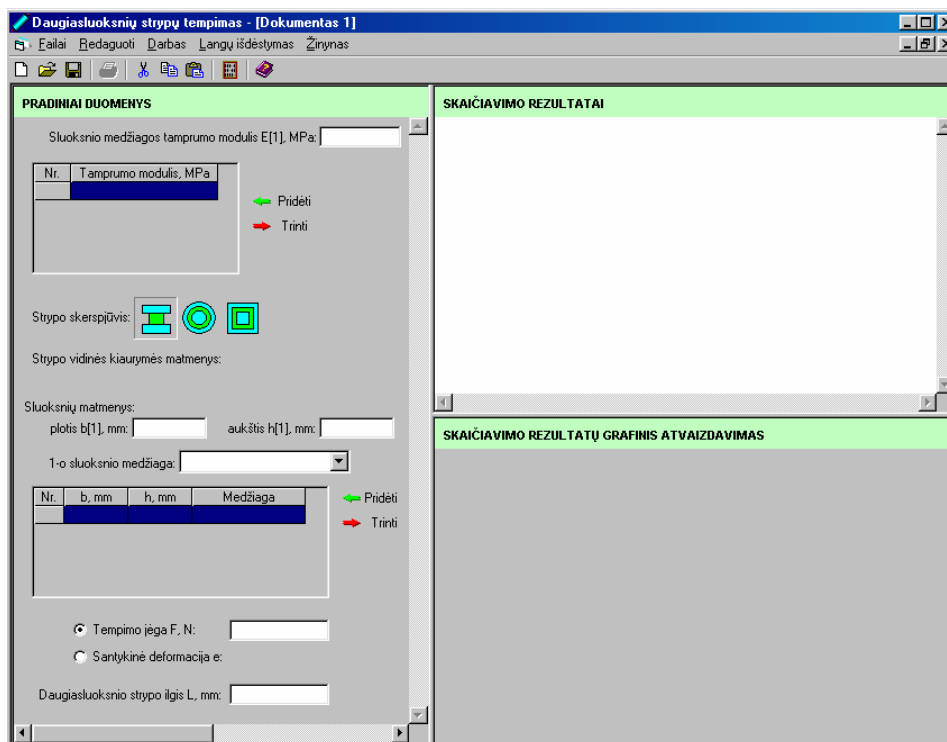
▪ Programos langas

Programos lange (13 pav.) yra šios dalys:

Pradiniai duomenys – duomenų įvedimo laukas;

Skaičiavimo rezultatai – pradinių duomenų kontrolės ir gautų rezultatų laukas;

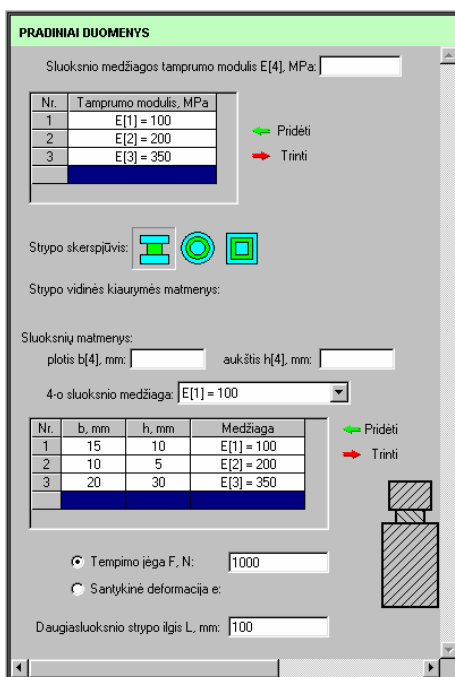
Skaičiavimo rezultatų grafinis atvaizdavimas – grafikų atvaizdavimo laukas.



13 pav. Programos langas

4.2.1. Programos lango dalis „Pradiniai duomenys“

Šioje lango dalyje (14 pav.) vartotojas gali įvesti pradinis duomenis: DKE sluoksnių tamprumo modulius, sluoksnių matmenis, priskirti DKE struktūrą, išsirinkti sluoksnių medžiagą, pasirinkti tempimo jėgą arba santykinę deformaciją.



14 pav. Pradiniai duomenys

Veiksmai atliekami programos pradinų duomenų lango dalyje:

- **Naujų duomenų įvedimas.** Pradinių duomenų įvedimui numatyti duomenų įvedimo laukeliai. Nauji duomenys įrašomi į duomenų įvedimo laukelį ir spaudžiant mygtuką „Pridėti“ arba klavišą „Enter“ duomenys patalpinami duomenų įvedimo lentelėje. Pavyzdžiui norint įvesti „Sluoksnio medžiagos tamprumo modulį E “ turi būti atlikti šie veiksmai: sluoksnio medžiagos tamprumo modulio vertė įrašoma į jai skirtą įvedimo laukelį ir spaudžiamas mygtukas „Pridėti“ arba klavišas „Enter“. Tamprumo modulio vertė bus įrašyta į duomenų lentelę. Norint įvesti sekančią sluoksnio tamprumo modulio vertę, reikia atlikti analogišką veiksmą. Tamprumo modulio vertė bus patalpinta sekančioje lentelės eilutėje ir t.t. Skaičius, esantis įrašė „Sluoksnio medžiagos tamprumo modulis $E[1]$ “ rodo, kelinta jo vertė patalpinta duomenų lentelėje. Visus lentelėje esančius įrašus galima peržiūrėti. Pereiti iš vieno duomenų įvedimo laukelio į kitą galima spaudžiant klavišą „Tab“.

- **Duomenų pašalinimas.** Jei duomenys suvesti neteisingai, tai spaudžiant mygtuką „Trinti“ pastarieji pašalinami. Šis veiksmas atliekamas sekančiai: duomenų lentelėje kairiuoju pelės klavišu pažymimas „Sluoksnio medžiagos tamprumo modulis $E[n]$ “ (pažymėta eilutė nulispalvina mėlyna spalva) ir spaudžiamas kairysis pelės klavišas ties „Trinti“. Neteisingi duomenys bus pašalinti iš duomenų lentelės.

Norint ištrinti duomenis duomenų įvedimo laukelyje, pavyzdžiui ištrinti „sluoksnio plotį“, jis pažymimas ir spaudžiamas klavišas „Delete“. Norint ištrinti duomenis, kurie yra būtini užduoties sprendimui, gaunamas pranešimas apie negalimą duomenų ištrynimą.

- **Keisti turimus duomenis.** Vartotojas nereikalingus arba klaidingai įvestus duomenys nebūtinai turi pašalinti, bet gali juos pakeisti naujais. Turimų duomenų keitimas atliekamas taip: kairiuoju pelės klavišu duomenų lentelėje pažymimas norimas keisti „Sluoksnio medžiagos tamprumo modulis $E[n]$ “. Pažymėta tamprumo modulio vertė atsiranda duomenų įvedimo laukelyje, o pažymėjus ją ir vietoj jos įrašius naują vertę spaudžiant klavišą „Enter“, nauja vertė pakeis duomenų lentelėje pažymėtą tamprumo modulio vertę.


- **Strypo skerspjūvio pasirinkimas.** Strypo skerspjūvio struktūrą pasirenkama spaudžiant kairįjį pelės klavišą ant reikiamos skerspjūvio struktūros.

- **Išsirinkti sluoksnio medžiagą.** Naują sluoksnio medžiagą galime išsirinkti iš medžiagų sąrašo. Čia naudojamas „1-o sluoksnio medžiagos“ pasirinkimo laukelis. Jo dešinėje yra mygtukas su trikampio formos ženklu, kurį paspaudus, sąrašas išsiskleidžia ir iš jo galima išsirinkti pageidaujamą sluoksnio medžiagą.

„Sluoksnio medžiagos“ sąrašo dydis priklauso nuo „Sluoksnio tamprumo modulio“ duomenų lentelėje įvestų įrašų skaičiaus. Skaičius, esantis šalia užrašo, pvz. „1 sluoksnio medžiaga“, nurodo kelinta sluoksnio medžiaga yra įrašoma į duomenų lentelę.

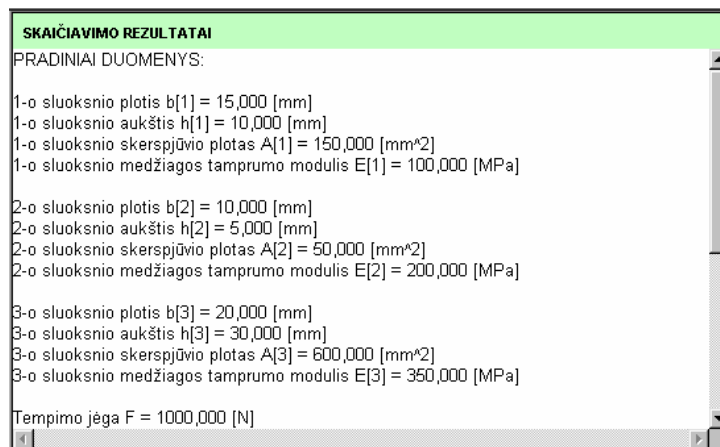
Programos lango „Pradiniai duomenys“ aprašymą galima iškviešti sužadinus meniu punktą „Pagalba“.

4.2.2. Programos lango dalis „Skaičiavimo rezultatai“

Įvedus pradinis užduoties duomenis, skaičiavimai atliekami pasirinkus priemonių juostoje esanti mygtuką  arba meniu punktą „Darbas“ ir iš papildomo meniu išsirinkus komandą „Skaičiuoti“.

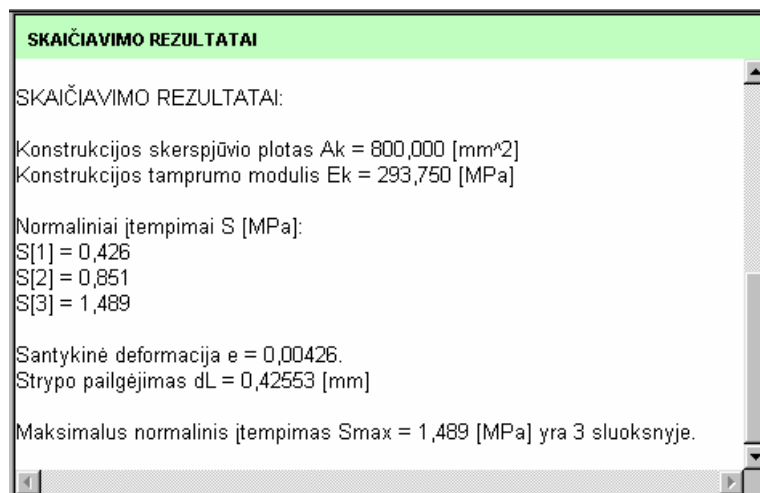
Skaičiavimo rezultatai atvaizduojami „Skaičiavimo rezultatai“ dalyje. Čia esanti informacija susideda iš dviejų dalių:

- pradiniai duomenys – matomi pirmoje dalyje (15a pav.). Toks pradinių duomenų atvaizdavimas leidžia vartotojui patikrinti jų teisingumą, t.y. ar įvesti duomenys atitinka užduoties sąlygą;



15a pav. Pradiniai duomenys

- skaičiavimo rezultatai pateikiami antroje dalyje (15b pav.).



15b pav. Skaičiavimo rezultatai

▪ **Rezultatų spausdinimas.** Rezultatų spausdinimas atliekamas taip: pelės žymeklis pastatomas dalyje „Skaiciavimo rezultatai“, spaudžiamas meniu punktas „Failai“ ir iš papildomo meniu išsirenkama komanda „Spausdinti“ arba priemonių juostoje spaudžiamas mygtukas su spausdintuvo piktograma. Atsidarius pasirinkimo langui, pasirenkama spausdinti skaičiavimo rezultatus ir/arba spausdinti grafikus.

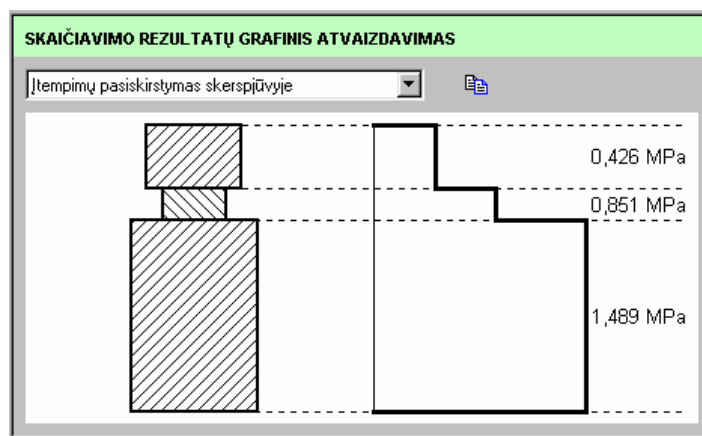
▪ **Rezultatų kopijavimas.** Norint kopijuoti skaičiavimo rezultatus reikia atlikti tokius veiksmus: jei reikia kopijuoti visus „Skaiciavimo rezultatai“ dalyje esančius duomenis (tiek pradinius, tiek skaičiavimo), pelės žymeklis pastatomas šioje dalyje, spaudžiamas meniu punktas „Redaguoti“ ir iš papildomo meniu išrenkama komanda „Kopijuoti“ arba priemonių kopijavimui skirtas mygtukas. Tuo atveju, jei reikia kopijuoti tik dalį rezultatų, juos reikia pažymėti kairiuoju pelės klavišu ir pasirinkti anksčiau minėtas komandas.

Programos lango „Skaiciavimo rezultatai“ aprašymą galima iškviešti sužadinus meniu punktą „Pagalba“.

4.2.3. Programos lango dalis „Skaiciavimo rezultatų grafinis atvaizdavimas“

Gautų skaičiavimo rezultatų grafinis vaizdas pateikiamas „Skaiciavimo rezultatų grafinis atvaizdavimas“ lango dalyje (16 pav.), naudojant pasirinkimo laukelį, kurio dešinėje yra mygtukas su trikampio formos ženklu. Pastarąjį paspaudus, sąrašas išsiskleidžia ir iš jo galima išsirinkti pageidaujamą grafiką.

1. DKE struktūra ir įtempimų pasiskirstymas sluoksniuose (viename grafike);
2. DKE tamprumo modulio priklausomybės nuo kiekvieno sluoksnio skerspjūvio ploto (viename grafike);
3. DKE stiprumo priklausomybės nuo kiekvieno sluoksnio skerspjūvio ploto (viename grafike).



16 pav. Skaičiavimo rezultatų grafinis atvaizdavimas

- **Rezultatų spausdinimas.**

- Rezultatų spausdinimas atliekamas spaudžiant meniu punktą „Failai“ ir iš papildomo meniu išsirenkant komandą „Spausdinti“ arba priemonių juostoje spaudžiant mygtuką su spausdintuvo piktograma, o atsidarius pasirinkimo langui, pasirenkama spausdinti skaičiavimo rezultatus ir/arba spausdinti grafikus.

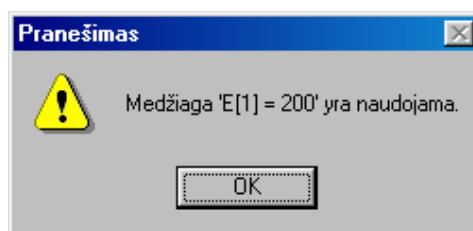
- **Rezultatų kopijavimas.**

Norint kopijuoti skaičiavimo rezultatų grafiką, spaudžiamas „Skaičiavimo rezultatų grafinis atvaizdavimas“ dalyje esantis mygtukas.

Programos lango „Skaičiavimo rezultatų grafinis atvaizdavimas“ aprašymą galima iškviešti sužadinus meniu punktą „Pagalba“.

4.2.4. Programos pranešimai

Programai dirbant vaizduoklyje gali atsirasti pranešimų, kurie perspėja vartotoją apie neteisingus jo veiksmus.



17 pav. Pranešimas „Medžiaga $E[1] = 200$ yra naudojama“

Programa gali pateikti šiuos pranešimus (klaidas):

1. „Medžiagų tamprumo modulis E turi būti teigiamas ir didesnis už 0“ – pakeisti arba įvesti naują vertę;
2. „Sluoksnio plotis b negali būti mažesnis arba lygus 0“ – pakeisti arba įvesti naują vertę;
3. „Sluoksnio aukštis h negali būti mažesnis arba lygus 0“ - pakeisti arba įvesti naują vertę;
4. „Sluoksnio storis t negali būti mažesnis arba lygus 0“ – įvesti teigiamą skaičių;
5. „Sluoksnio storis x ašies kryptimi db negali būti mažesnis arba lygus 0“ - įvesti teigiamą skaičių;
6. „Sluoksnio storis y ašies kryptimi db negali būti mažesnis arba lygus 0“ - įvesti teigiamą skaičių;
7. „Vidinis spindulys $R(0)$ negali būti mažesnis už 0“ - įvesti teigiamą spindulio vertę;

8. „Vidinis strypo plotis $b[0]$ negali būti mažesnis už 0!“ - įvesti teigiamą spindulio vertę;
9. „Vidinis strypo aukštis $h(0)$ negali būti mažesnis už 0!“ - įvesti naują vertę arba pataisyti turimą;
10. „Tempimo jėga F turi būti didesnė už 0!“ – įvesti teigiamą tempimo jėgos vertę;
11. „Nėra nei vieno sluoksnio! Įvesti bent vieną sluoksnį“ –pasirinkti sluoksnį iš sąrašo;
12. „Prašome pasirinkti sluoksnio medžiagą“ - pasirinkti sluoksnio medžiagą iš sąrašo;
13. „Strypo deformacija e turi būti didesnė už 0“ - įvesti teigiamą deformacijos vertę;
14. „ Strypo ilgis L turi būti didesnis už 0“ – įvesti strypo vertę.

4.3. Programos instaliavimo dokumentas

Kompiuteryje, kuriame bus naudojama programa, turi būti įdiegta nuo Windows 98 iki Windows XP operacinė aplinka.

Programa vartotojui pateikiama kompaktiniame diske. Kompiuterio laikmenoje C:\, sukuriama programai skirtas darbinis aplankalas „Tempimas“. Iš kompaktinio disko į sukurtą aplankalą perkopijuojami tame diske esantys failai.

Programa paleidžiama suaktyvinus **C:\Tempimas\DKE.exe**. Jei programa startavo sėkmingai, vaizduoklyje pasirodo „pasirinkimų“ langas (11 pav.).

Spaudžiant mygtuką „Pradėti darbą“ atsiranda pagrindinis programos langas - tai sėkmingas programos starto kriterijus.

4.3.1. Reikalinga techninė įranga

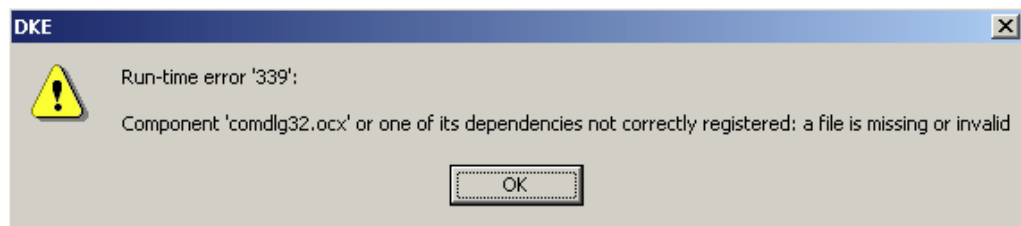
Reikalavimas kompiuteriams, kuriuose galima įdiegti programą:

- Sparta: 200 MHz ir greitesni;
- Operatyvioji atmintis: mažiausiai 64 RAM (rekomenduojama 128 MB);
- Laikmenos talpa: mažiausiai 20 GB;
- Išorinis nuskaitymo įrenginys: CD-ROM;
- Pelė, klaviatūra, 15” monitorius.

4.4. Sistemos administratoriaus vadovas

Kompiuteriuose, kuriuose naudojama programa turi būti įdiegta operacinė aplinka Windows 98, 2000, ME, XP.

Jei programa, nestartuoja, t.y. indikuojama klaida, pranešimu Run-time error '339', tai reiškia, kad operacinėje aplinkoje nėra MSHFLXGD.OCX ir RICHTX32.OCX failų, tuo atveju, būtina failus įdiegti.



18 pav. Pranešimas

5. PRODUKTO KOKYBĖS ĮVERTINIMAS

Sukurtą programą numatoma įdiegti KTU Panevėžio instituto Technologijos fakultete. Projektuojant šią programą, nuolat buvo konsultuojamasi su šio fakulteto dėstytojais, t.y. dėstytojais vedančiais medžiagų mechanikos modulį, doktorantais, mokslininkais, dirbančiais daugiasluoksnių konstrukcijų srityje.

Programa buvo išbandyta dalyvaujant Mechanikos inžinerijos katedros dėstytojams. Išklausius kiekvieno minėto darbuotojo pastabas, pageidavimus, buvo padaryti kai kurie pataisymai.

Programa sudomino dėstytojus, nes per gana trumpą laiką galima atlikti skaičiavimus ir nubraižyti grafikus, kurie atskleidžia dalyko esmę.

Atlikus apklausą paaiškėjo, kad programa tenkina vartotoją. Be to darbuotojai galėjo patys išbandyti programą, išspręsti kelias užduotis, kurios paprastai pateikiamos studentams. Dėstytojai pareiškė, kad programa patikima, lengvai įsisavinama, pateikia pakankamai tikslūs skaičiavimo rezultatus. Todėl, jų nuomone, sukurtą programą bus galima naudoti studentų praktinių užduočių atlikimui.

6. IŠVADOS

1. Pristatomoje programoje panaudotas daugiasluoksnių konstrukcinių elementų (DKE) parametrų skaičiavimo algoritmas, realizuojantis idealizuotą DKE modelį.
2. Pasirinktas vienmatis ašinio tempimo (tik tamprumo zonoje) DKE modelis, kuris būdamas paprastu yra pakankamai adekvatus.
3. Pristatoma programa – mokomoji, todėl toks idealizacijos lygmens pasirinkimas supaprastino algoritmą kartu atspindėdamas esminius daugiasluoksnių konstrukcinių elementų parametrų skaičiavimo metodų ypatumus.
4. Programa palyginta su panašia užsienyje sukurta programa, skaičiuojančia dviejų medžiagų kompozitą tamprioje zonoje.
5. Kuriant pristatomą programą dėl pateikimo metodikos buvo konsultuotasi su KTU Panevėžio instituto dėstytojais, nes programa – mokomoji.
6. Nustatytas minimalus pradinių duomenų skaičius, įgalinantis suskaičiuoti visus DKE mechaninius parametrus, būtinus DKE projektavimui.
7. Programa skaičiuoja tris dažniausiai naudojamas daugiasluoksnio struktūras.
8. Sukurtas algoritmas, kurį realizuoja pristatoma programa, suprojektuota ir suprogramuota grafinė vartotojo sąsaja.
9. Skaičiavimams sukurta originali taikomoji programa firmos „Microsoft“ objektinėje programavimo kalboje Visual Basic 6, veikianti Windows 9x/ME/2000/XP operacinėse aplinkose.
10. Pristatomoji programa pateikia pakankamai tikslus skaičiavimo ir grafinius rezultatus.

7. LITERATŪRA

1. Hodges H. Technology in Ancient World, Pelikan Books, 1971. - p.562
2. Schartz M.M. Composite Materials, Volume II. Processing, Fabrication and Applications, New Jersey, 1997. - p.568.
3. A.Žiliukas, A.Keršys, N.Meslinas. Estimation Criteria of Elasto-Plastic Behavior of Composites // Mechanika. 2003. Nr.2(40). p.21-24.
4. J.Bareišis, V.Kleiza. Calculation of stretched multilayer bars under elastic and plastoelastic deformation // Proceedings of international conference Mechanika-2001. Kaunas: KTU, Technologija, 2001. p.254-259.
5. V.Kleiza. Tampriai plastinio uždavinio sprendimas komutuojančių matricų erdvėje // Lietuvos matematikos rinkinys. ISSN:0132-2818. Vilnius: TEV, 2001, nr. spec.nr.(41). p. 511-516.
6. <http://callisto.my.mtu.edu/my4150/class1./class1.html>
7. Ostreika A. Programavimo Visual Basic pagrindai: mokomoji knyga.- Kaunas: KTU, Technologija, 2003. 225 p.
8. Starkus B. Visual Basic 6 Jūsų kompiuteryje. K., 2000. 284 p.
9. Bareišis J. Medžiagų mechanika: vadovėlis. – Šiauliai: Šiaulių universiteto leidykla, 2000.
10. Bareišis J., Paulauskas V. Daugiasluoksnių kompozicinių konstrukcinių elementų projektavimas. – Kaunas: KTU, Technologija, 1995 – 48 p.
11. Bareišis J., Tričys V. Polimerų ir kompozitų mechanika. – Šiauliai: Šiaulių universiteto leidykla, 2003. – 390 p.

8. The Modelling of Design Process of the Stretched Multilayer Structural Elements (Summary)

Summary

This thesis presents the software program for multilayer structural elements design. The main users of the software should be bachelor and master students of mechanical technology department and their professors as well. The program can be successfully applied for computations of mechanical characteristics of the real (industrial) problems. This program is able to choose the multilayer structural elements type and its mechanical and geometrical parameters. While the program is running the user can watch the calculated values of the parameters, plots of the main dependencies and explanations of main operations being performed with the program variables. The algorithm of the program assumes that the model multilayer structural elements is idealized, i.e. uniaxial stretching, which is simple and quite adequate. The program is developed for training purposes, so the algorithm used is idealized and not sophisticated but preserving the main features of multilayer structural elements calculation.

There is given a comparison of the program with the similar one calculating two materials composite developed by foreign researchers.

The program calculates three most often used multilayer structural elements structures (there is no limit of the amount of layers). The presented original program is written in Microsoft programming language Visual Basic 6 and can be used on computers running Windows 9x/Me/Xp.

9. TERMINŲ IR SANTRAUPŲ ŽODYNAS

DKE – daugiasluoksnis konstrukcinis elementas.

Ašinis tempimas – tai toks apkrovimo būdas, kai strypą veikiančių jėgų atstojamosios pridėjimo taškas sutampa su skerspjūvio geometriniu centru.

Deformacija – tai strypo absoliutaus pailgėjimo ir pradinio (nedeformuoto) strypo ilgio santykis

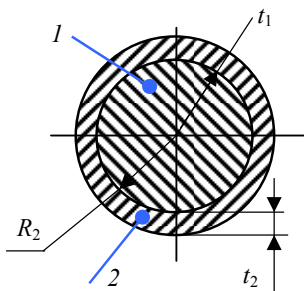
Tamprumo modulis – proporcingumo koeficientas tiesinėje įtempimo ir deformacijos priklausomybėje

10. PRIEDAI

1 priedas

Daugiasluoksnių strypų skaičiavimo užduotys

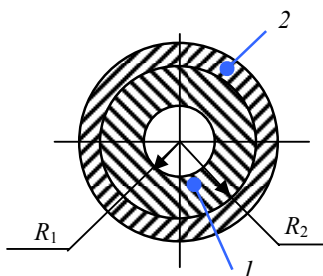
1. Užduotis



19 pav.

Rasti cilindrinio tempiamo strypo, kurio skerspjūvis parodytas 19 pav., įtempimus vidiniame strype 1 bei išoriniame strype 2, kai apkrova $F = 10^5$ N, $R_1 = 0$, $t_1 = 15$ mm, $R_2 = 15$ mm, $t_2 = 5$ mm $E_1 = 1,2 \cdot 10^5$ MPa (varis) ir $E_2 = 2 \cdot 10^5$ MPa (plienas). Apskaičiuoti strypo ekvivalentinį tamprumo modulį E_K ir strypo absoliutinį pailgėjimą, jei $l = 3 \cdot 10^3$ mm.

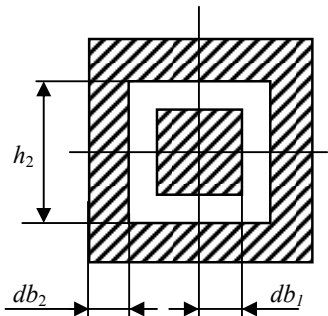
2. Užduotis



20 pav.

Sukonstruoti dvisluksnį cilindrinį strypą, kurio skerspjūvis parodytas 20 pav., sudarytą iš stikloplastčio ($R_1 = 20$ mm, $t_1 = 5$ mm $E_1 = 9 \cdot 10^3$ MPa) ir epoksidinės dervos ($E_2 = 2 \cdot 10^3$ MPa). Tegul tokį strypą veikia 150000 N tempimo jėga. Parinkti tokį epoksidinės dervos skerspjūvio plotą, kad įtempimas stikloplastyje neviršytų 200 MPa vertės.

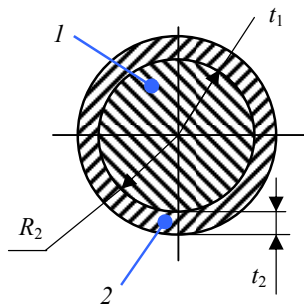
3. Užduotis



21 pav.

Rasti kvadratinio dvisluksnio tempiamo strypo įtempimus vidiniame strype 1, kai apkrova $F = 10^5$ N, $h_1 = b_1 = 0$ mm, $dh_1 = db_1 = 10$ mm, $E_1 = 1,2 \cdot 10^5$ MPa (varis) ir $h_2 = b_2 = 30$ mm, $dh_2 = db_2 = 5$ mm, $E_2 = 2 \cdot 10^5$ MPa (plienas). Apskaičiuoti strypo ekvivalentinį tamprumo modulį E_K ir jo deformaciją

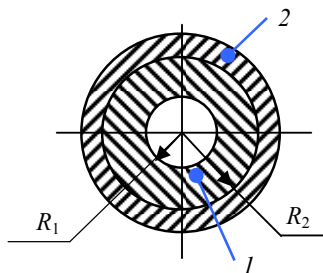
4. Užduotis



22 pav.

Rasti cilindrinio tempiamo strypo, kurio skerspjūvis parodytas 22 pav., įtempimus vidiniame strype 1 bei išoriniame strype 2, esant jo deformacijai $\varepsilon = 0,87$, jei $R_1 = 0$, $t_1 = 15$ mm, $R_2 = 15$ mm, $t_2 = 25$ mm $E_1 = 1,2 \cdot 10^5$ MPa (varis) ir $E_2 = 2 \cdot 10^5$ MPa (plienas). Suskaičiuoti strypo apkrovą, kuri sukėlė duotą deformaciją.

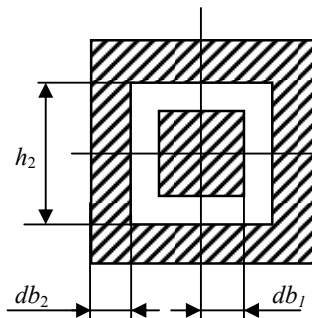
5. Užduotis



23 pav.

Sukonstruoti dvisluoksnį cilindrinį strypą, kurio skerspjūvis parodytas 23 pav., sudarytą iš stikloplasčio ($R_1 = 20$ mm, $t_1 = 5$ mm $E_1 = 9 \cdot 10^3$ MPa) ir epoksidinės dervos ($E_2 = 2 \cdot 10^3$ MPa). Tegul tokį strypo deformacijos vertė $\varepsilon = 1,07$. Parinkti tokį epoksidinės dervos skerspjūvio plotą, kad įtempimas stikloplastyje neviršytų 200 MPa vertės.

6. Užduotis



24 pav.

Rasti kvadratinio dvisluoksniu tempiamo strypo įtempimus vidiniame strype 1, esant jo deformacijai $\varepsilon = 2,64$ ir sekantiems skerspjūvio parametrams: $h_1 = b_1 = 0$ mm, $dh_1 = db_1 = 10$ mm, $E_1 = 1,2 \cdot 10^5$ MPa (varis) ir $h_2 = b_2 = 30$ mm, $dh_2 = db_2 = 5$ mm, $E_2 = 2 \cdot 10^5$ MPa (plienas). Suskaičiuoti strypo ekvivalentinį tamprumo modulį E_K ir strypo absoliutinį pailgėjimą.

7 lentelė. Medžiagos tamprumo moduliai

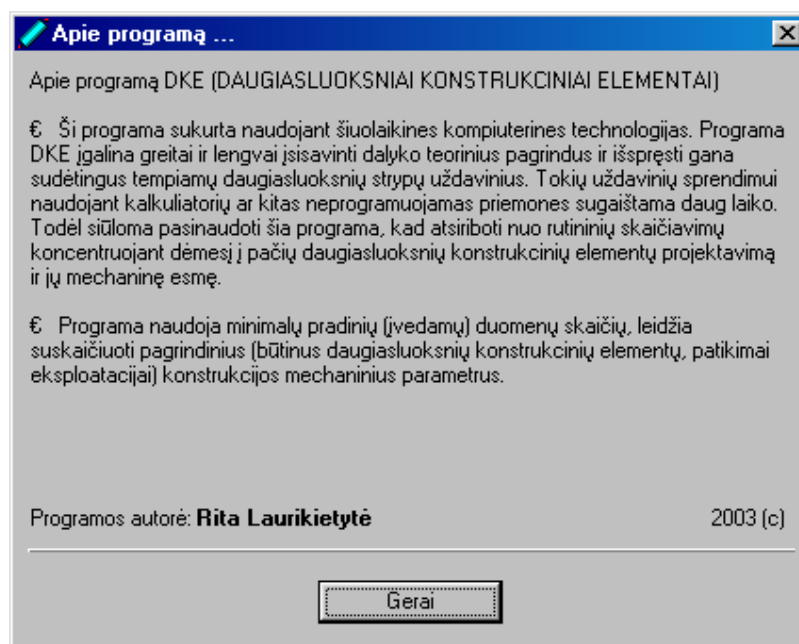
Medžiaga	Tamprumo modulis E , 10^4 MPa
Medis	1,02
Stiklas	7,0
Švinas	1,6
Aliuminis	7,1
Diuraliuminis	7,3
Marmuras	5,6
Granitas	4,9
Varis	12,3
Sidabras	7,9
Žalvaris	9,8
Ketus	10,8
Plienas	20,0
Organinis stiklas	0,32
Polistirolas	0,32
Volframas	39,0
Indis	52,8
Kapronas	70

A N K E T A

Jūsų pateikta informacija padės nustatyti programos, skirtos tempiamų daugiasluoksnių strypų mechaninių parametrų skaičiavimui, reikalingumą.

1. Jūsų pareigos:
 Profesorius;
 Docentas;
 Doktorantas;
 Lektorius;
 Studentas.
2. Ar Jus tenkina tokia programa:
 Taip;
 Ne.
3. Jei netenkina, tai kodėl:
 Nemokate dirbti su kompiuteriu;
 Galvojate, kad studentai mažiau dirbs savarankiškai;
 Esate prie to pripratę ir keisti nieko nenorite;
Kita .
4. Ar ši programa yra patikima:
 Taip;
 Ne.
5. Jei nepatikima tai kodėl:
 Aiškiau, kai skaičiuojama kalkuliatoriumi;
 Nepasitikite kompiuteriu;
Kita .
6. Ar patogiu naudotis vartotojo sąsaja:
 Taip;
 Ne.
7. Ar sutiktumėte instaliuoti tokią programą:
 Taip;
 Ne.

Dėkoju už atsakymus



25 pav. Apie programą

Programos kodai

```

Public fMainForm As frmMain
Public frmApie As frmAbout

Public Type Stac
    x As Double
    y As Double
    dx As Double
    dy As Double
End Type

Sub Main()
    frmPradinis.Show
    Set fMainForm = New frmMain
    Set frmApie = New frmAbout
    Load fMainForm
End Sub

Public Function GStac(x As Double, y As Double, dx As Double, dy As Double) As Stac
    Dim s As Stac
    s.x = x
    s.y = y
    s.dx = dx
    s.dy = dy
    GStac = s
End Function

' gr - kur grafika atvaizduoti
' pavDydis - grafiko lango dydis
' Tsk(Sk) - tasku skaicius grafike
' Rodyti(Sk) - ar rodyti grafika
' x(Sk,Tsk) - grafiku x koordiantes
' y(Sk,Tsk) - grafiku y koordiantes
' Sp(Sk) - grafiku spalvos
' St(Sk) - grafiku liniju storiai
' Tip(Sk) - grafiku liniju tipai
' Pav(Sk) - kiekvieno grafiko zymejimas
' RodytiPav - nusako ar vaizduoti grafiku pavadinimus
' Xpav - x asies dydio zymejimas
' Ypav - x asies dydio zymejimas
Public Sub Grafikai(gR As PictureBox, pavDydis As Stac, _
    Tsk() As Integer, Rodyti() As Boolean, _
    x() As Double, y() As Double, Sp() As Long, _
    St() As Integer, Tip() As Integer, _
    Pav() As String, Optional RodytiPav As Boolean = True, _
    Optional Xpav As String = "", _
    Optional Ypav As String = "")
    ' Grafiku skaicius
    Dim Sk, Sk1 As Integer
    Sk = UBound(x, 1)
    Sk1 = UBound(y, 1)
    If Sk1 < Sk Then Sk = Sk1
    ' Grafiku sumazejimas del uzrasu (pixel'iais)
    Dim skX, skX1, skY, skY1, zym As Integer
    skX = 0: skX1 = 0: skY = 0: skY1 = 0: zym = 0
    ' Apskaiciuoju max ir min grafiko reiksmes
    Dim Xmin As Double, Ymin As Double, _
        Xmax As Double, Ymax As Double
    Xmin = x(0, 0): Ymin = y(0, 0): Xmax = x(0, 0): Ymax = y(0, 0)
    For I = 0 To Sk
        If Rodyti(I) Then
            For j = 0 To Tsk(I) - 1
                If x(I, j) > Xmax Then
                    Xmax = x(I, j)
                Else
                    If x(I, j) < Xmin Then Xmin = x(I, j)
                End If
                If y(I, j) > Ymax Then
                    Ymax = y(I, j)
                Else
                    If y(I, j) < Ymin Then Ymin = y(I, j)
                End If
            Next j
        End If
    Next I

```

```

' Perskaiciuoju min ir max reikšmes nes braizomas tinklelis
Dim Nx As Integer, Ny As Integer, _
    kiekX As Integer, kiekY As Integer
Dim dx As Double, dy As Double ' tinklelio zingsniai
Dim xaf As String, yaf As String
Nx = 5
Ny = 5
MinMaxRibos Xmin, Xmax, Nx, dx, kiekX
MinMaxRibos Ymin, Ymax, Ny, dy, kiekY
' Asiu skaiciu formatas
If kiekX = 0 Then
    xaf = "0"
Else
    xaf = "0."
    For I = 0 To kiekX - 1
        xaf = xaf + "0"
    Next I
End If
If kiekY = 0 Then
    yaf = "0"
Else
    yaf = "0."
    For I = 0 To kiekY - 1
        yaf = yaf + "0"
    Next I
End If
' Nustatau reikiama vietos dydi asiu zymejimui
Const dyy As Integer = 50 ' skaiciu atstumas nuo y asies
Const dxx As Integer = 50 ' skaiciu atstumas nuo x asies
Const ZymKreivIlgis As Integer = 300 ' Liniju ilgis zymejime
td = gR.TextWidth(Ypav)
If td > skX Then skX = td
For I = 0 To Ny
    yy = Ymin + I * dy
    td = gR.TextWidth(Format(yy, yaf))
    If td > skX Then skX = td
Next I
skX = skX + dyy
skY1 = gR.TextHeight(Format(yy, yaf)) / 2
td = gR.TextHeight(Xpav)
If td > skY Then skY = td
For I = 0 To Nx
    xx = Xmin + I * dx
    td = gR.TextHeight(Format(xx, xaf))
    If td > skY Then skY = td
Next I
skY = skY + dxx
skX1 = gR.TextWidth(Format(xx, xaf)) / 2
' Nustatau reikiama vietos dydi grafiku zymejimui atvaizduoti
If RodytiPav Then
    For I = 0 To Sk
        If Rodyti(I) Then
            td = gR.TextWidth(Pav(I)) + ZymKreivIlgis + dxx
            If td > zym Then zym = td
        End If
    Next I
End If
' Jei neužduoti grafiko gabaritai, tai juos apskaičiuoju
If pavDydis.dx = 0 Then pavDydis.dx = gR.ScaleWidth - 2 * pavDydis.x
If pavDydis.dy = 0 Then pavDydis.dy = gR.ScaleHeight - 2 * pavDydis.y
' Išvalau grafiku braizymo erdve
gR.FillStyle = vbFSSolid
gR.FillColor = gR.BackColor
gR.ForeColor = gR.BackColor
gR.Line (-1, -1)-(gR.ScaleWidth + 1, gR.ScaleHeight + 1), , B
' Apskaiciuoju x ir y asiu mastelius
Dim mX, mY As Double
mX = (pavDydis.dx - skX - skX1 - zym) / Abs(Xmax - Xmin)
mY = (pavDydis.dy - skY - skY1) / Abs(Ymax - Ymin)
' Apskaiciuoju koordinaciu pradios taskus
Dim X0, Y0 As Long
X0 = pavDydis.x + skX - mX * Xmin
Y0 = pavDydis.y + skY1 + mY * Ymax
' Braizau tinkleli
gR.ForeColor = RGB(0, 0, 0)
gR.DrawStyle = vbSolid
gR.DrawWidth = 1
yy1 = Y0 - mY * Ymin 'pavDydis.y
yy2 = Y0 - mY * Ymax 'pavDydis.y + tdy * Ny - skY

```



```

For I = 0 To Nx
    xx = X0 + mX * (Xmin + I * dx)
    gR.Line (xx, yy1)-(xx, yy2)
Next I
xx1 = X0 + mX * Xmin 'pavDydis.x + skX
xx2 = X0 + mX * Xmax 'xx1 + tdx * Nx
For I = 0 To Ny
    yy = Y0 - mY * (Ymin + I * dy) 'pavDydis.y + i * tdy
    gR.Line (xx1, yy)-(xx2, yy)
Next I
' Braizau grafikus
Dim PixTiwp As Integer
yy1 = gR.CurrentY
gR.ScaleMode = vbPixels
gR.CurrentY = gR.CurrentY + 1
gR.ScaleMode = vbTwips
PixTiwp = Abs(gR.CurrentY - yy1)
For I = 0 To Sk
    If Rodyti(I) Then
        gR.DrawWidth = St(I)
        gR.DrawStyle = Tip(I)
        gR.ForeColor = Sp(I)
        For j = 0 To Tsk(I) - 2
            If Tip(I) <> vbSolid Then
                gR.DrawWidth = 1
                For k = 0 To St(I) - 1
                    gR.Line (X0 + mX * x(I, j), Y0 - mY * y(I, j) + k * PixTiwp)- _
                        (X0 + mX * x(I, j + 1), Y0 - mY * y(I, j + 1) + k * PixTiwp)
                Next k
            Else
                gR.Line (X0 + mX * x(I, j), Y0 - mY * y(I, j))- _
                    (X0 + mX * x(I, j + 1), Y0 - mY * y(I, j + 1))
            End If
        Next j
    End If
Next I
' Isvadame skaičius ant asių
gR.ForeColor = RGB(0, 0, 0)
xx1 = X0 + mX * Xmin - dxx
For I = 0 To Ny
    yy = Ymin + I * dy
    yy1 = Y0 - mY * yy
    If I = Ny - 1 Then
        yys = Ypav
    Else
        yys = Format(yy, yaf)
    End If
    td1 = gR.TextWidth(yys)
    td2 = gR.TextHeight(yys)
    gR.CurrentX = xx1 - td1
    gR.CurrentY = yy1 - td2 / 2
    gR.Print yys
Next I
yy1 = Y0 - mY * Ymin + dyy
For I = 0 To Nx
    xx = Xmin + I * dx
    xx1 = X0 + mX * xx
    If I = Nx - 1 Then
        xxs = Xpav
    Else
        xxs = Format(xx, xaf)
    End If
    td1 = gR.TextWidth(xxs)
    gR.CurrentX = xx1 - td1 / 2
    gR.CurrentY = yy1
    gR.Print xxs
Next I
' Isvadame grafiku zymejima
xx1 = X0 + mX * Xmax + skX1 + dxx
xx3 = xx1 + ZymKreivIlgis
xx2 = xx3 - dxx
yy1 = pavDydis.y
If RodytiPav Then
    For I = 0 To Sk
        If Rodyti(I) Then
            gR.DrawWidth = St(I)
            gR.DrawStyle = Tip(I)
            gR.ForeColor = Sp(I)
            yy3 = gR.TextHeight(Pav(I))

```

```

yy2 = yy1 + yy3 / 2
If Tip(I) <> vbSolid Then
    gR.DrawWidth = 1
    For k = 0 To St(I) - 1
        gR.Line (xx1, yy2 + k * PixTiwp)-(xx2, yy2 + k * PixTiwp)
    Next k
Else
    gR.Line (xx1, yy2 + k * PixTiwp)-(xx2, yy2 + k * PixTiwp)
End If
gR.CurrentX = xx3
gR.CurrentY = yy1
gR.ForeColor = 0
gR.Print Pav(I)
yy1 = yy1 + yy3 + dyy
End If
Next I
End If
End Sub

```

```

Public Function Apvalinti(ka As Double, kiek As Integer, Optional proc As Double = 10) As Double
    Dim xxx, k1 As Double
    Dim Sk As Integer
    xxx = ka * proc / 100#
    k1 = Round(1# / xxx)
    If k1 = 0 Then
        kiek = 0
        skp = 0
        While (ka / 10 ^ skp) > 10
            skp = skp + 1
        Wend
        Sk = Round(ka / 10 ^ skp)
        Apvalinti = Sk * 10 ^ skp
    Else
        kiek = 0
        While CInt(CInt(10 ^ kiek) / CInt(k1)) = 0
            kiek = kiek + 1
        Wend
        xxx = Round(ka, kiek)
        Apvalinti = xxx
    End If
End Function

```

' Prskaiciuoja minimalia, maksimalia ribas, intervalo dydi ir intervalu
' skaiciu taip, kad sie dydziai butu 'grazus', t.y. su ribotu skaiciumi
' po kablelio.

```

Public Sub MinMaxRibos(MinSk As Double, MaxSk As Double, N As Integer, _
    dSk As Double, Optional kiek As Integer)
    If Abs(MaxSk - MinSk) <= 0.0000000000001 Then
        MinSk = MinSk - 0.5
        MaxSk = MaxSk + 0.5
    End If
    dSk = Apvalinti(Abs(MaxSk - MinSk) / N, (kiek))
    I = 1
    aMin = Abs(MinSk)
    While dSk * I < aMin
        I = I + 1
    Wend
    If MinSk > 0 Then MinSk = dSk * (I - 1)
    If MinSk < 0 Then MinSk = -dSk * I
    N = 1
    While MinSk + dSk * N < MaxSk
        N = N + 1
    Wend
    If 100 * Abs(MinSk + dSk * (N - 1) - MaxSk) / MaxSk < 0.1 Then N = N - 1
    MaxSk = MinSk + N * dSk
End Sub

```

```

Public YraSkaiciavimoRezultatai As Boolean

Dim Vx, Vy As Integer
Dim RedaguotiM As Boolean, RedaguotiS As Boolean

Dim Rv() As Double ' Sluoksniu vidiniai spinduliai
Dim Ri() As Double ' Sluoksniu isoriniai spinduliai
Dim bv() As Double ' Sluoksniu vidiniai plociai
Dim bi() As Double ' Sluoksniu isoriniai plociai
Dim hv() As Double ' Sluoksniu vidiniai auksciai
Dim hi() As Double ' Sluoksniu isoriniai auksciai
Dim A() As Double ' Sluoksniu skerspjuviu plotai

```

```

Dim E() As Double ' Sluoksniu medziagos tamprumo moduliai
Dim s() As Double ' Itempimu dydis sluoksniuose
Dim MaxS As Double ' Maksimalus itempimai
Dim MaxSsluoksnis As Integer
Dim Ssk As Integer ' Sluoksniu skaicius
Dim Ak As Double ' Strypo skerspjuvio plotas
Dim Ek As Double ' Bendras strypo tamprumo modulis
Dim Bk As Double ' Strypo standumas tempimui
Dim Sk As Double ' Bendras strypo itempimas  $Sk=N/Ak$ 
Dim N As Double ' Strypo tempimo jega
Dim eps As Double ' Strypo deformacija
Dim Ls As Double ' Strypo ilgis

' Duomenys grafiku braizymui
Dim gR() As Boolean ' ar rodyti grafika
Dim gTsk() As Integer ' tasku skaicius grafike
Dim gA() As Double ' x koordinates
Dim gEk() As Double ' y koordinates
Dim gBk() As Double ' y koordinate
Dim gSp() As Long ' grafiku liniju spalva
Dim gSt() As Integer ' grafiku liniju storiai
Dim gTip() As Integer ' grafiku liniju tipai
Dim gPavEk() As String ' grafiku kreiviu pavadinimai
Dim gPavBk() As String ' grafiku kreiviu pavadinimai

Private Sub Combo1_Click()
    If MSHFlexGrid2.Row <> MSHFlexGrid2.Rows - 1 Then
        If Toolbar2.Buttons.Item(1).Value = tbrPressed Or Toolbar2.Buttons.Item(3).Value = tbrPressed Then
            MSHFlexGrid2.Col = 3
            MSHFlexGrid2.Text = Combo1.Text
        End If
        If Toolbar2.Buttons.Item(2).Value = tbrPressed Then
            MSHFlexGrid2.Col = 2
            MSHFlexGrid2.Text = Combo1.Text
        End If
    End If
    YraSkaiciavimoRezultatai = False
    RichTextBox1.Text = ""
    Combo2.Visible = False
    Combo3.Visible = False
    Graf.Visible = False
    Toolbar4.Visible = False
    Meniu_Mygtukai_TextBox
End Sub

Private Sub Combo2_Click()
    If Combo2.Visible = True Then
        Select Case Combo2.ListIndex
            Case 0
                Combo3.Visible = False
                Graf.Top = Combo2.Top + Combo2.Height + Graf.Left
            Case 1
                Graf.Top = Combo3.Top + Combo3.Height + Graf.Left
                Combo3.Visible = True
                Combo3.Clear
                Combo3.AddItem "Ek priklausomybė nuo visų plotų"
                For I = 0 To Ssk
                    Combo3.AddItem "Ek priklausomybė nuo ploto A[" + CStr(I + 1) + "]"
                Next I
                Combo3.ListIndex = 0
            Case 2
                Graf.Top = Combo3.Top + Combo3.Height + Graf.Left
                Combo3.Visible = True
                Combo3.Clear
                Combo3.AddItem "Bk priklausomybė nuo visų plotų"
                For I = 0 To Ssk
                    Combo3.AddItem "Bk priklausomybė nuo ploto A[" + CStr(I + 1) + "]"
                Next I
                Combo3.ListIndex = 0
        End Select
        Form_Resize
        GrafineDalis
    End If
End Sub

Private Sub Combo3_Click()
    GrafineDalis
End Sub

```

```

Public Sub IsrinktiEilute(ByVal kas As MSHFlexGrid, ByVal eNr As Integer)
    kas.Row = eNr
    kas.RowSel = eNr
    kas.Col = 0
    kas.ColSel = kas.Cols - 1
End Sub

```

```

Private Sub Form_Load()
    YraSkaiciavimoRezultatai = False
    PrDuum.Top = 0
    PrDuum.Left = 0
    Picture1.Top = 400
    Picture1.Left = 0
    Rezult.Top = 0
    RichTextBox1.Top = 400
    RichTextBox1.Left = 0
    Picture3.Top = 400
    Picture3.Left = 0
    VScroll1.Top = 0
    HScroll1.Left = 0
    Vx = 0
    Vy = 0
    RedaguotiM = False
    RedaguotiS = False

    MSHFlexGrid1.Row = 0
    MSHFlexGrid1.Col = 0
    MSHFlexGrid1.Text = "Nr."
    MSHFlexGrid1.Col = 1
    MSHFlexGrid1.Text = "Tamprumo modulis, MPa"
    For I = 0 To 1
        MSHFlexGrid1.Col = I
        MSHFlexGrid1.CellAlignment = flexAlignCenterCenter
    Next
    MSHFlexGrid1.ColWidth(0) = 500
    MSHFlexGrid1.ColWidth(1) = 1970
    MSHFlexGrid1.RowHeightMin = MSHFlexGrid1.CellHeight
    MSHFlexGrid2.RowHeightMin = MSHFlexGrid2.CellHeight
    MSHFlexGrid1.Row = 1

    Combo1.Clear
    SluoksniuLentele1

    IsrinktiEilute MSHFlexGrid1, 1
    IsrinktiEilute MSHFlexGrid2, 1

    Combo2.Clear
    Combo2.AddItem "Itempimų pasiskirstymas skerspjuvyje"
    Combo2.AddItem "Ek priklausomybė nuo sluoksnių skerspjuvio ploto"
    Combo2.AddItem "Bk priklausomybė nuo sluoksnių skerspjuvio ploto"
    Combo2.ListIndex = 0
    Graf.Left = 120
    Graf.Top = Combo2.Top + Combo2.Height + Graf.Left
End Sub

```

```

Private Sub Form_Resize()
    On Error Resume Next
    Const Tarpas As Integer = 15
    x = 4 * Me.ScaleWidth / 9
    y = Me.ScaleHeight / 2

    PrDuum.Width = x - Tarpas
    PrDuum.Height = Me.ScaleHeight
    Picture1.Width = PrDuum.ScaleWidth
    Picture1.Height = PrDuum.ScaleHeight - Picture1.Top

    Rezult.Left = x + Tarpas
    Rezult.Width = Me.ScaleWidth - PrDuum.Width - Tarpas
    Rezult.Height = y - Tarpas
    RichTextBox1.Width = Rezult.ScaleWidth
    RichTextBox1.Height = Rezult.ScaleHeight - RichTextBox1.Top

    GrafLan.Left = x + Tarpas
    GrafLan.Top = y + Tarpas
    GrafLan.Width = Me.ScaleWidth - PrDuum.Width - Tarpas
    GrafLan.Height = y - Tarpas
    Picture3.Width = GrafLan.ScaleWidth
    Picture3.Height = GrafLan.ScaleHeight - Picture3.Top
    Graf.Width = Picture3.ScaleWidth - 2 * Graf.Left

```

```

Graf.Height = Picture3.ScaleHeight - Graf.Top - Graf.Left
If Combo2.Visible = True Then Graf.Invalidate

VScroll1.Height = Picture1.ScaleHeight - HScroll1.Height
VScroll1.Left = Picture1.ScaleWidth - VScroll1.Width
vkiek = PrDuumLaukas.Height + HScroll1.Height - Picture1.ScaleHeight
VScroll1.Max = vkiek
If vkiek <= 0 Then
    VScroll1.Enabled = False
Else
    VScroll1.Enabled = True
End If
HScroll1.Width = Picture1.ScaleWidth - VScroll1.Width
HScroll1.Top = Picture1.ScaleHeight - HScroll1.Height
hkiek = PrDuumLaukas.Width + VScroll1.Width - Picture1.ScaleWidth
HScroll1.Max = hkiek
If hkiek <= 0 Then
    HScroll1.Enabled = False
Else
    HScroll1.Enabled = True
End If
End Sub

Private Sub SluoksniuLentele1()
    MSHFlexGrid2.Cols = 4
    MSHFlexGrid2.Rows = 2
    MSHFlexGrid2.Row = 0
    MSHFlexGrid2.Col = 0
    MSHFlexGrid2.Text = "Nr."
    MSHFlexGrid2.Col = 1
    MSHFlexGrid2.Text = "b, mm"
    MSHFlexGrid2.Col = 2
    MSHFlexGrid2.Text = "h, mm"
    MSHFlexGrid2.Col = 3
    MSHFlexGrid2.Text = "Medžiaga"
    For I = 0 To 3
        MSHFlexGrid2.Col = I
        MSHFlexGrid2.CellAlignment = flexAlignCenterCenter
    Next
    MSHFlexGrid2.Row = 1
    For I = 0 To 3
        MSHFlexGrid2.Col = I
        MSHFlexGrid2.Text = ""
    Next
    MSHFlexGrid2.Row = 1
    MSHFlexGrid2.ColWidth(0) = 400
    MSHFlexGrid2.ColWidth(1) = 850
    MSHFlexGrid2.ColWidth(2) = 850
    MSHFlexGrid2.ColWidth(3) = 1520
End Sub

Private Sub SluoksniuLentele2()
    MSHFlexGrid2.Cols = 3
    MSHFlexGrid2.Rows = 2
    MSHFlexGrid2.Row = 0
    MSHFlexGrid2.Col = 0
    MSHFlexGrid2.Text = "Nr."
    MSHFlexGrid2.Col = 1
    MSHFlexGrid2.Text = "t, mm"
    MSHFlexGrid2.Col = 2
    MSHFlexGrid2.Text = "Medžiaga"
    For I = 0 To 2
        MSHFlexGrid2.Col = I
        MSHFlexGrid2.CellAlignment = flexAlignCenterCenter
    Next
    MSHFlexGrid2.Row = 1
    For I = 0 To 2
        MSHFlexGrid2.Col = I
        MSHFlexGrid2.Text = ""
    Next
    MSHFlexGrid2.Row = 1
    MSHFlexGrid2.ColWidth(0) = 400
    MSHFlexGrid2.ColWidth(1) = 850
    MSHFlexGrid2.ColWidth(2) = 2370
End Sub

Private Sub SluoksniuLentele3()
    MSHFlexGrid2.Cols = 4
    MSHFlexGrid2.Rows = 2

```

```

MSHFlexGrid2.Row = 0
MSHFlexGrid2.Col = 0
MSHFlexGrid2.Text = "Nr."
MSHFlexGrid2.Col = 1
MSHFlexGrid2.Text = "db, mm"
MSHFlexGrid2.Col = 2
MSHFlexGrid2.Text = "dh, mm"
MSHFlexGrid2.Col = 3
MSHFlexGrid2.Text = "Medžiaga"
For I = 0 To 3
    MSHFlexGrid2.Col = I
    MSHFlexGrid2.CellAlignment = flexAlignCenterCenter
Next
MSHFlexGrid2.Row = 1
For I = 0 To 3
    MSHFlexGrid2.Col = I
    MSHFlexGrid2.Text = ""
Next
MSHFlexGrid2.Row = 1
MSHFlexGrid2.ColWidth(0) = 400
MSHFlexGrid2.ColWidth(1) = 850
MSHFlexGrid2.ColWidth(2) = 850
MSHFlexGrid2.ColWidth(3) = 1520
End Sub

' Lygina TextBox'o reiksme su 0 pagal uzduota salyga.
Public Function TestText(txt As TextBox, Zenklas As String, ByVal Pranesimas As String) As Boolean
    TestText = True
    Dim skaicius As String
    skaicius = Replace(txt.Text, ",", ".")
    Sk = Val(skaicius)
    Select Case Zenklas
        Case ">"
            If Sk > 0 Then TestText = False
        Case ">="
            If Sk >= 0 Then TestText = False
        Case "<"
            If Sk < 0 Then TestText = False
        Case "<="
            If Sk <= 0 Then TestText = False
        Case "="
            If Sk = 0 Then TestText = False
    End Select
    If (Not TestText) Or (IsNumeric(skaicius) = False) Then
        TestText = False
        kas = MsgBox(Pranesimas, vbOKOnly + vbExclamation, "Pranesimas")
        txt.SetFocus
    End If
    txt.Text = skaicius
End Function

Private Sub Form_Unload(Cancel As Integer)
    fMainForm.AtidarytaLangu = fMainForm.AtidarytaLangu - 1
    If fMainForm.AtidarytaLangu = 0 Then
        fMainForm.mnuFileSave.Enabled = False
        fMainForm.mnuFileClose.Enabled = False
        fMainForm.mnuFilePrint.Enabled = False
        fMainForm.mnuEditCut.Enabled = False
        fMainForm.mnuEditCopy.Enabled = False
        fMainForm.mnuEditPaste.Enabled = False
        fMainForm.mnuDarbasSkaiciuoti.Enabled = False
        fMainForm.tbToolBar.Buttons.Item(3).Enabled = False ' Saugoti
        fMainForm.tbToolBar.Buttons.Item(4).Enabled = False ' Uzduoti
        fMainForm.tbToolBar.Buttons.Item(6).Enabled = False ' Spausdinti
        fMainForm.tbToolBar.Buttons.Item(8).Enabled = False ' Iskirpti
        fMainForm.tbToolBar.Buttons.Item(9).Enabled = False ' Kopijuoti
        fMainForm.tbToolBar.Buttons.Item(10).Enabled = False ' Istatyti
        fMainForm.tbToolBar.Buttons.Item(12).Enabled = False ' Skaiciuoti
    End If
End Sub

Private Sub HScroll1_Change()
    PrDuumLaukas.Move PrDuumLaukas.Left + Vx - HScroll1.Value, PrDuumLaukas.Top, PrDuumLaukas.Width, PrDuumLaukas.Height
    Vx = HScroll1.Value
End Sub

Private Sub HScroll1_Scroll()

```

```

HScroll1_Change
End Sub
Private Sub TextGotFocus(txt As TextBox)
    txt.SelStart = 0
    txt.SelLength = Len(txt.Text)
    txt.Tag = 1
End Sub

Private Sub Label4_Click()

End Sub

Private Sub MSHFlexGrid1_Click()
    MSHFlexGrid1.Row = MSHFlexGrid1.RowSel
    MSHFlexGrid1.Col = 1
    Text1.Text = TamprumoModulis(MSHFlexGrid1.Text)
    MedziagosNumeris MSHFlexGrid1.RowSel
    Meniu_Mygtukai_TextBox
    IsrinktiEilute MSHFlexGrid1, MSHFlexGrid1.RowSel
    If MSHFlexGrid1.RowSel = MSHFlexGrid1.Rows - 1 Then
        RedaguotiM = False
    Else
        RedaguotiM = True
    End If
End Sub

Private Sub MedziagosNumeris(ByVal Nr As Integer)
    Label4.Caption = "Sluoksniu medziagos tamprumo modulis E[" & Nr & "], MPa: "
End Sub

Public Function TamprumoModulis(Pav As String) As Double
    If Pav = "" Then
        TamprumoModulis = 0
        Exit Function
    End If
    Dim I As Integer: I = 1
    While (Mid(Pav, I, 1) <> "=") Or (I > Len(Pav))
        I = I + 1
    Wend
    TamprumoModulis = Val(Replace(Mid(Pav, I + 1, Len(Pav)), ",", "."))
End Function

Private Sub MSHFlexGrid2_Click()
    MSHFlexGrid2.Row = MSHFlexGrid2.RowSel
    MSHFlexGrid2.Col = 1
    Text4.Text = MSHFlexGrid2.Text
    If Toolbar2.Buttons.Item(1).Value = tbrPressed Or Toolbar2.Buttons.Item(3).Value = tbrPressed Then
        MSHFlexGrid2.Col = 2
        Text5.Text = MSHFlexGrid2.Text
        MSHFlexGrid2.Col = 3
    Else
        MSHFlexGrid2.Col = 2
    End If
    If MSHFlexGrid2.Row <> MSHFlexGrid2.Rows - 1 Then
        For I = 0 To Combo1.ListCount
            If Combo1.List(I) = MSHFlexGrid2.Text Then
                Combo1.ListIndex = I
                Exit For
            End If
        Next
    End If
    Meniu_Mygtukai_TextBox
    SluoksniuNumeris MSHFlexGrid2.RowSel
    IsrinktiEilute MSHFlexGrid2, MSHFlexGrid2.RowSel
    If MSHFlexGrid2.RowSel = MSHFlexGrid2.Rows - 1 Then
        RedaguotiS = False
    Else
        RedaguotiS = True
    End If
End Sub

Private Sub Option1_Click()
    If Text6.Top <> 6700 Then
        Text6.Top = 6700
        Text6.Text = ""
    End If
End Sub

Private Sub Option2_Click()

```

```

If Text6.Top <> 7050 Then
    Text6.Top = 7050
    Text6.Text = ""
End If
End Sub

Private Sub RichTextBox1_GotFocus()
    Meniu_Mygtukai_RichTextBox
End Sub

Private Sub Text1_GotFocus()
    TextGotFocus Text1
    Meniu_Mygtukai_TextBox
End Sub

Private Sub Text1_LostFocus()
    Text1.Tag = 0
End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        If TestText(Text1, "<=", "Medžiagos tamprumo modulis E turi būti teigiamas ir didesnis už 0!") Then
            If Not RedaguotiM Then
                PridetiMedziaga
            Else
                MSHFlexGrid1.Row = MSHFlexGrid1.RowSel
                MSHFlexGrid1.Col = 1
                SenaMedz = MSHFlexGrid1.Text
                MSHFlexGrid1.Text = MedziagosPavadinimas(Text1.Text, MSHFlexGrid1.Row)
                Nr = MSHFlexGrid2.Row
                If Toolbar2.Buttons.Item(1).Value = tbrPressed Or Toolbar2.Buttons.Item(3).Value = tbrPressed Then
                    MSHFlexGrid2.Col = 3
                Else
                    MSHFlexGrid2.Col = 2
                End If
                For j = 1 To MSHFlexGrid2.Rows - 2
                    MSHFlexGrid2.Row = j
                    If MSHFlexGrid2.Text = SenaMedz Then MSHFlexGrid2.Text = MSHFlexGrid1.Text
                Next
                IsrinktiEilute MSHFlexGrid2, Nr
                RedaguotiM = False
            End If
            IsrinktiEilute MSHFlexGrid1, MSHFlexGrid1.Rows - 1
            MedziagosNumeris MSHFlexGrid1.Rows - 1
            Text1.Text = ""
        End If
    End If
End Sub

Private Sub Text2_GotFocus()
    TextGotFocus Text2
    Meniu_Mygtukai_TextBox
End Sub

Private Sub Text2_LostFocus()
    Text2.Tag = 0
End Sub

Private Sub Text2_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        If TestText(Text2, "<", "Dydis turi būti teigiamas arba lygus 0!") Then
            YraSkaiciavimoRezultatai = False
            RichTextBox1.Text = ""
            Combo2.Visible = False
            Combo3.Visible = False
            Graf.Visible = False
            Toolbar4.Visible = False
            Skerspjuvis Picture2
        End If
    End If
End Sub

Private Sub Text3_GotFocus()
    TextGotFocus Text3
    Meniu_Mygtukai_TextBox
End Sub

Private Sub Text3_LostFocus()
    Text3.Tag = 0

```



```

End Sub

Private Sub Text3_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        If TestText(Text3, "<", "Dydis turi būti teigiamas arba lygus 0!") Then
            YraSkaiciavimoRezultatai = False
            RichTextBox1.Text = ""
            Combo2.Visible = False
            Combo3.Visible = False
            Graf.Visible = False
            Toolbar4.Visible = False
            Skerspjuvis Picture2
        End If
    End If
End Sub

Private Sub Text4_GotFocus()
    TextGotFocus Text4
    Meniu_Mygtukai_TextBox
End Sub

Private Sub Text4_LostFocus()
    Text4.Tag = 0
End Sub

Private Sub Text4_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 And MSHFlexGrid2.Row <> MSHFlexGrid2.Rows - 1 Then
        If TestText(Text4, "<=", "Dydis turi būti teigiamas ir didesnis už 0!") Then
            If Not RedaguotiS Then
                PridetiSluoksni
            Else
                Nr = MSHFlexGrid2.Row
                MSHFlexGrid2.Col = 1
                MSHFlexGrid2.Text = Val(Text4.Text)
                YraSkaiciavimoRezultatai = False
                RichTextBox1.Text = ""
                Combo2.Visible = False
                Combo3.Visible = False
                Graf.Visible = False
                Toolbar4.Visible = False
                Skerspjuvis Picture2
                RedaguotiS = False
            End If
            IsrinktiEilute MSHFlexGrid2, MSHFlexGrid2.Rows - 1
            SluoksniuNumeris MSHFlexGrid2.Rows - 1
            Text4.Text = ""
            Text5.Text = ""
        End If
    End If
End Sub

Private Sub Text5_GotFocus()
    TextGotFocus Text5
    Meniu_Mygtukai_TextBox
End Sub

Private Sub Text5_LostFocus()
    Text5.Tag = 0
End Sub

Private Sub Text5_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 And MSHFlexGrid2.Row <> MSHFlexGrid2.Rows - 1 Then
        If TestText(Text5, "<=", "Dydis turi būti teigiamas ir didesnis už 0!") Then
            If Not RedaguotiS Then
                PridetiSluoksni
            Else
                Nr = MSHFlexGrid2.Row
                MSHFlexGrid2.Col = 2
                MSHFlexGrid2.Text = Val(Text5.Text)
                YraSkaiciavimoRezultatai = False
                RichTextBox1.Text = ""
                Combo2.Visible = False
                Combo3.Visible = False
                Graf.Visible = False
                Toolbar4.Visible = False
                Skerspjuvis Picture2
                RedaguotiS = False
            End If
            IsrinktiEilute MSHFlexGrid2, MSHFlexGrid2.Rows - 1

```

```

        MedziagosNumeris MSHFlexGrid2.Rows - 1
        Text5.Text = ""
        Text4.Text = ""
    End If
End If
End Sub

Private Sub Text6_GotFocus()
    TextGotFocus Text6
    Meniu_Mygtukai_TextBox
End Sub

Private Sub Text6_LostFocus()
    Text6.Tag = 0
End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
    Select Case Button.Key
        Case "Pridėti"
            PridetiMedziaga
        Case "Trinti"
            AtimtiMedziaga
    End Select
    Meniu_Mygtukai_TextBox
End Sub

Private Sub PridetiMedziaga()
    If Not TestText(Text1, "<=", "Medziagos tamprumo modulis E turi būti teigiamas ir didesnis už 0!") Then Exit Sub

    MSHFlexGrid1.Row = MSHFlexGrid1.Rows - 1
    MSHFlexGrid1.Col = 0
    MSHFlexGrid1.Text = MSHFlexGrid1.Row
    MSHFlexGrid1.Col = 1
    MSHFlexGrid1.Text = MedziagosPavadinimas(Text1.Text, MSHFlexGrid1.Row)
    For I = 0 To 1
        MSHFlexGrid1.Col = I
        MSHFlexGrid1.CellAlignment = flexAlignCenterCenter
    Next
    MedziagosNumeris MSHFlexGrid1.Rows
    MSHFlexGrid1.Rows = MSHFlexGrid1.Rows + 1
    SluoksniuMedziagos
    'MSHFlexGrid1.Row = MSHFlexGrid1.Rows - 1
    MedziagosNumeris MSHFlexGrid1.Rows - 1
    IsrinktiEilute MSHFlexGrid1, MSHFlexGrid1.Rows - 1
    RedaguotiM = False
    Text1.SetFocus
    TextGotFocus Text1
End Sub

Private Sub AtimtiMedziaga()
    ' Tikrinu ar nera naudojama medziaga
    Dim I As Integer
    Dim NaudojamaMedz As Boolean
    NaudojamaMedz = False
    If Toolbar2.Buttons.Item(1).Value = tbrPressed Or Toolbar2.Buttons.Item(3).Value = tbrPressed Then
        MSHFlexGrid2.Col = 3
    Else
        MSHFlexGrid2.Col = 2
    End If
    MSHFlexGrid1.Col = 1
    MSHFlexGrid1.Row = MSHFlexGrid1.RowSel
    For I = 1 To MSHFlexGrid2.Rows - 2
        MSHFlexGrid2.Row = I
        If MSHFlexGrid2.Text = MSHFlexGrid1.Text Then
            NaudojamaMedz = True
            Exit For
        End If
    Next
    If NaudojamaMedz Then
        x = MsgBox("Medziaga "" & MSHFlexGrid1.Text & "" yra naudojama.", vbOKOnly + vbExclamation, "Pranešimas")
        Exit Sub
    End If
    ' Trinu medziaga
    kuris = MSHFlexGrid1.RowSel
    If MSHFlexGrid1.RowSel <> (MSHFlexGrid1.Rows - 1) Then
        MSHFlexGrid1.RemoveItem MSHFlexGrid1.RowSel
        kiek = MSHFlexGrid1.Rows - 2
        For I = 1 To kiek
            MSHFlexGrid1.Row = I

```

```

MSHFlexGrid1.Col = 0
MSHFlexGrid1.Text = I
MSHFlexGrid1.Col = 1
SenaMedz = MSHFlexGrid1.Text
MSHFlexGrid1.Text = MedziagosPavadinimas(TamprumoModulis(MSHFlexGrid1.Text), I)
If Toolbar2.Buttons.Item(1).Value = tbrPressed Or Toolbar2.Buttons.Item(3).Value = tbrPressed Then
    MSHFlexGrid2.Col = 3
Else
    MSHFlexGrid2.Col = 2
End If
For j = 1 To MSHFlexGrid2.Rows - 2
    MSHFlexGrid2.Row = j
    If MSHFlexGrid2.Text = SenaMedz Then MSHFlexGrid2.Text = MSHFlexGrid1.Text
Next
Next
SluoksniuMedziagos
MSHFlexGrid1.Row = kuris
MSHFlexGrid1.Col = 1
Text1.Text = TamprumoModulis(MSHFlexGrid1.Text)
MedziagosNumeris kuris
IsrinktiEilute MSHFlexGrid1, kuris
If kuris = MSHFlexGrid1.Rows - 1 Then
    RedaguotiM = False
Else
    RedaguotiM = True
End If
End If
End Sub

Private Function MedziagosPavadinimas(ByVal txt As String, ByVal Nr As Integer) As String
    MedziagosPavadinimas = "E[" & Nr & "]" = " & txt
End Function

Public Sub SluoksniuMedziagos()
    kurim = Combo1.ListIndex
    If kurim < 0 Then kurim = 0
    Combo1.Clear
    MSHFlexGrid1.Col = 1
    Dim I As Integer
    For I = 1 To MSHFlexGrid1.Rows - 2
        MSHFlexGrid1.Row = I
        Combo1.AddItem MSHFlexGrid1.Text
    Next
    If kurim > Combo1.ListCount - 1 Then kurim = Combo1.ListCount - 1
    Combo1.ListIndex = kurim
End Sub

Public Sub Toolbar2_ButtonClick(ByVal Button As MSComctlLib.Button)
    If Button.Tag = 1 Then
        Select Case Button.Key
            Case "Pirmas"
                Toolbar2.Buttons.Item(1).Value = tbrPressed
            Case "Antras"
                Toolbar2.Buttons.Item(2).Value = tbrPressed
            Case "Trecias"
                Toolbar2.Buttons.Item(3).Value = tbrPressed
        End Select
    Exit Sub
End If
Select Case Button.Key
    Case "Pirmas"
        Toolbar2.Buttons.Item(1).Value = tbrPressed
        Toolbar2.Buttons.Item(1).Tag = 1
        Toolbar2.Buttons.Item(2).Value = tbrUnpressed
        Toolbar2.Buttons.Item(2).Tag = 0
        Toolbar2.Buttons.Item(3).Value = tbrUnpressed
        Toolbar2.Buttons.Item(3).Tag = 0
    Case "Antras"
        Toolbar2.Buttons.Item(2).Value = tbrPressed
        Toolbar2.Buttons.Item(2).Tag = 1
        Toolbar2.Buttons.Item(1).Value = tbrUnpressed
        Toolbar2.Buttons.Item(1).Tag = 0
        Toolbar2.Buttons.Item(3).Value = tbrUnpressed
        Toolbar2.Buttons.Item(3).Tag = 0
    Case "Trecias"
        Toolbar2.Buttons.Item(3).Value = tbrPressed
        Toolbar2.Buttons.Item(3).Tag = 1
        Toolbar2.Buttons.Item(1).Value = tbrUnpressed
        Toolbar2.Buttons.Item(1).Tag = 0

```

```

        Toolbar2.Buttons.Item(2).Value = tbrUnpressed
        Toolbar2.Buttons.Item(2).Tag = 0
    End Select
    Toolbar2.Refresh
    If Button.Key = "Pirmas" And Button.Value = tbrPressed Then
        SluoksniuLentele1
        Label8.Visible = False
        Text2.Visible = False
        Label9.Visible = False
        Text3.Visible = False
        Label11.Caption = "plotis b[1], mm: "
        Label12.Caption = "aukštis h[1], mm: "
        Label12.Visible = True
        Text4.Text = ""
        Text5.Text = ""
        Text5.Visible = True
    End If
    If Button.Key = "Antras" And Button.Value = tbrPressed Then
        SluoksniuLentele2
        Label8.Caption = "spindulys R[0], mm: "
        Label8.Visible = True
        Text2.Text = ""
        Text2.Visible = True
        Label9.Visible = False
        Text3.Visible = False
        Label11.Caption = "storis t[1], mm: "
        Label12.Visible = False
        Text4.Text = ""
        Text5.Visible = False
    End If
    If Button.Key = "Trecias" And Button.Value = tbrPressed Then
        SluoksniuLentele3
        Label8.Caption = "plotis b[0], mm: "
        Label8.Visible = True
        Text2.Text = ""
        Text2.Visible = Visible
        Label9.Caption = "aukštis h[0], mm: "
        Label9.Visible = True
        Text3.Text = ""
        Text3.Visible = Visible
        Label11.Caption = "plotis db[1], mm: "
        Label12.Caption = "aukštis dh[1], mm: "
        Label12.Visible = True
        Text4.Text = ""
        Text5.Text = ""
        Text5.Visible = True
    End If
    Meniu_Mygtukai_TextBox
    Skerspjuvis Picture2
    SluoksniuNumeris MSHFlexGrid2.Rows - 1
    IsrinktiEilute MSHFlexGrid2, MSHFlexGrid2.Rows - 1
End Sub

Private Sub Toolbar3_ButtonClick(ByVal Button As MSCometLib.Button)
    Select Case Button.Key
        Case "Pridėti"
            PridetiSluoksni
        Case "Trinti"
            TrintiSluoksni
    End Select
    kuris = MSHFlexGrid2.RowSel
    RichTextBox1.Text = ""
    Combo2.Visible = False
    Combo3.Visible = False
    Graf.Visible = False
    Meniu_Mygtukai_TextBox
    Skerspjuvis Picture2
    IsrinktiEilute MSHFlexGrid2, kuris
End Sub

Private Sub PridetiSluoksni()
    If Toolbar2.Buttons.Item(1).Value = tbrPressed Then
        If Not TestText(Text4, "<", "Sluoksniu plotis b negali būti mažesnis arba lygus 0.") Then Exit Sub
        If Not TestText(Text5, "<", "Sluoksniu auštis h negali būti mažesnis arba lygus 0.") Then Exit Sub
        If Combo1.Text = "" Then
            kaip = MsgBox("Prašome parinkti sluoksniu medžiagą.", vbOKOnly + vbExclamation, "Pranešimas")
            Combo1.SetFocus
        End If
    End If
End Sub

```

```

MSHFlexGrid2.Row = MSHFlexGrid2.Rows - 1
MSHFlexGrid2.Col = 0
MSHFlexGrid2.Text = MSHFlexGrid2.Row
MSHFlexGrid2.Col = 1
MSHFlexGrid2.Text = Replace(Text4.Text, ",", ".")
MSHFlexGrid2.Col = 2
MSHFlexGrid2.Text = Replace(Text5.Text, ",", ".")
MSHFlexGrid2.Col = 3
MSHFlexGrid2.Text = Combo1.Text
For I = 0 To 3
    MSHFlexGrid2.Col = I
    MSHFlexGrid2.CellAlignment = flexAlignCenterCenter
Next
End If
If Toolbar2.Buttons.Item(2).Value = tbrPressed Then
If Not TestText(Text4, "<", "Sluoksnio storis t negali būti mažesnis arba lygus 0.") Then Exit Sub
If Combo1.Text = "" Then
    kaip = MsgBox("Prašome parinkti sluoksnio medžiagą.", vbOKOnly + vbExclamation, "Pranešimas")
    Combo1.SetFocus
    Exit Sub
End If
MSHFlexGrid2.Row = MSHFlexGrid2.Rows - 1
MSHFlexGrid2.Col = 0
MSHFlexGrid2.Text = MSHFlexGrid2.Row
MSHFlexGrid2.Col = 1
MSHFlexGrid2.Text = Replace(Text4.Text, ",", ".")
MSHFlexGrid2.Col = 2
MSHFlexGrid2.Text = Combo1.Text
For I = 0 To 2
    MSHFlexGrid2.Col = I
    MSHFlexGrid2.CellAlignment = flexAlignCenterCenter
Next
End If
If Toolbar2.Buttons.Item(3).Value = tbrPressed Then
If Not TestText(Text4, "<", "Sluoksnio storis x ašies kryptimi db negali būti mažesnis arba lygus 0.") Then Exit Sub
If Not TestText(Text5, "<", "Sluoksnio storis y ašies kryptimi dh negali būti mažesnis arba lygus 0.") Then Exit Sub
If Combo1.Text = "" Then
    kaip = MsgBox("Prašome parinkti sluoksnio medžiagą.", vbOKOnly + vbExclamation, "Pranešimas")
    Combo1.SetFocus
    Exit Sub
End If
MSHFlexGrid2.Row = MSHFlexGrid2.Rows - 1
MSHFlexGrid2.Col = 0
MSHFlexGrid2.Text = MSHFlexGrid2.Row
MSHFlexGrid2.Col = 1
MSHFlexGrid2.Text = Replace(Text4.Text, ",", ".")
MSHFlexGrid2.Col = 2
MSHFlexGrid2.Text = Replace(Text5.Text, ",", ".")
MSHFlexGrid2.Col = 3
MSHFlexGrid2.Text = Combo1.Text
For I = 0 To 3
    MSHFlexGrid2.Col = I
    MSHFlexGrid2.CellAlignment = flexAlignCenterCenter
Next
End If
MSHFlexGrid2.Rows = MSHFlexGrid2.Rows + 1
SluoksnioNumeris MSHFlexGrid2.Rows - 1
IsrinktiEilute MSHFlexGrid2, MSHFlexGrid2.Rows - 1
RedaguotiS = False
Text4.SetFocus
End Sub

Private Sub TrintiSluoksn()
' kuris = MSHFlexGrid2.RowSel
If MSHFlexGrid2.RowSel <> (MSHFlexGrid2.Rows - 1) Then
    kuris = MSHFlexGrid2.RowSel
    MSHFlexGrid2.RemoveItem kuris
    For I = 1 To MSHFlexGrid2.Rows - 2
        MSHFlexGrid2.Row = I
        MSHFlexGrid2.Col = 0
        MSHFlexGrid2.Text = I
    Next
    If kuris = MSHFlexGrid2.Rows - 1 Then
        RedaguotiS = False
    Else
        RedaguotiS = True
    End If
End If
' IsrinktiEilute MSHFlexGrid2, kuris

```

```

End Sub

Private Sub SluoksniuNumeris(Nr As Integer)
    If Toolbar2.Buttons.Item(1).Value = tbrPressed Then
        Label11.Caption = "plotis b[" & Nr & "], mm: "
        Label12.Caption = "aukštis h[" & Nr & "], mm: "
    End If
    If Toolbar2.Buttons.Item(2).Value = tbrPressed Then
        Label11.Caption = "storis t[" & Nr & "], mm: "
    End If
    If Toolbar2.Buttons.Item(3).Value = tbrPressed Then
        Label11.Caption = "plotis db[" & Nr & "], mm: "
        Label12.Caption = "aukštis dh[" & Nr & "], mm: "
    End If
    Label5.Caption = Nr & "-o sluoksniu medžiaga: "
End Sub

Private Sub Toolbar4_ButtonClick(ByVal Button As MSCometLib.Button)
    Select Case Button.Key
        Case "Kopijuoti"
            Clipboard.Clear
            Clipboard.SetData Graf.Image
    End Select
End Sub

Private Sub VScroll1_Change()
    PrDuumLaukas.Move PrDuumLaukas.Left, PrDuumLaukas.Top + Vy - VScroll1.Value, PrDuumLaukas.Width, PrDuumLaukas.Height
    Vy = VScroll1.Value
End Sub

Private Sub VScroll1_Scroll()
    VScroll1_Change
End Sub

Public Function stof(s As String) As String
    ss = Val(Replace(s, ",", "."))
    sks = CStr(ss)
    stof = Replace(sks, ",", ".")
End Function

Public Sub SaugotiPradiniusDuomenisFaile(FailoVardas As String)
    On Error Resume Next
    Open FailoVardas For Output As #1
    ' Medžiagu saugojimas
    Dim z As Integer: z = MSHFlexGrid1.Rows - 2

    Print #1, z
    MSHFlexGrid1.Col = 1
    For j = 1 To z
        MSHFlexGrid1.Row = j
        Print #1, stof(TamprumoModulis(MSHFlexGrid1.Text))
    Next
    ' Skerspjūvio tipo saugojimas
    Dim StSk As Integer
    If Toolbar2.Buttons.Item(1).Value = tbrPressed Then
        StSk = 3
        Print #1, 1
    End If
    If Toolbar2.Buttons.Item(2).Value = tbrPressed Then
        StSk = 2
        Print #1, 2
        Print #1, stof(Text2.Text)
    End If
    If Toolbar2.Buttons.Item(3).Value = tbrPressed Then
        StSk = 3
        Print #1, 3
        Print #1, stof(Text2.Text)
        Print #1, stof(Text3.Text)
    End If
    ' Sluoksniu saugojimas
    z = MSHFlexGrid2.Rows - 2
    Print #1, z
    For l = 1 To z
        MSHFlexGrid2.Row = l
        For j = 1 To StSk
            MSHFlexGrid2.Col = j
            If j = StSk Then
                Skaic = Mid(MSHFlexGrid2.Text, 3)
                Print #1, stof(Skaic)
            End If
        Next j
    Next l
End Sub

```

```

        Else
            Print #1, MSHFlexGrid2.Text
        End If
    Next
Next
' Apkrovos saugojimas
kas = 0
If Option1.Value = True Then kas = 1
Print #1, kas
Print #1, stofs(Text6.Text)
Print #1, stofs(Text7.Text)
Close #1
End Sub

Public Function NuskaitytiPradiniusDuomenisIsFailo(FailoVardas As String) As Boolean
    On Error Resume Next
    NuskaitytiPradiniusDuomenisIsFailo = True
    Dim z As Integer, Sk As Double, ee As String, StSk As Integer
    Open FailoVardas For Input As #1
    ' Medziagu nuskaitymas
    Combo1.Clear
    Line Input #1, ee
    If Not IsNumeric(ee) Or EOF(1) Then
        NuskaitytiPradiniusDuomenisIsFailo = False
        Close #1
        Exit Function
    End If
    z = Val(ee)
    For l = 1 To z
        MSHFlexGrid1.Row = l
        MSHFlexGrid1.Col = 0
        MSHFlexGrid1.Text = l
        MSHFlexGrid1.Col = 1
        Line Input #1, ee
        If Not IsNumeric(ee) Or EOF(1) Then
            NuskaitytiPradiniusDuomenisIsFailo = False
            Close #1
            Exit Function
        End If
        MSHFlexGrid1.Text = MedziagosPavadinimas(ee, l)
        For k = 0 To 1
            MSHFlexGrid1.Col = k
            MSHFlexGrid1.CellAlignment = flexAlignCenterCenter
        Next
        MSHFlexGrid1.Rows = MSHFlexGrid1.Rows + 1
    Next
    MedziaguSkaicius = z
    SluoksniuMedziagos
    ' Nuskaitau skerspjuvio tipa
    Line Input #1, ee
    If Not IsNumeric(ee) Or EOF(1) Then
        NuskaitytiPradiniusDuomenisIsFailo = False
        Close #1
        Exit Function
    End If
    z = Val(ee)
    If (z < 1) Or (z > 3) Then
        NuskaitytiPradiniusDuomenisIsFailo = False
        Close #1
        Exit Function
    End If
    If z = 1 Then
        Toolbar2_ButtonClick Toolbar2.Buttons.Item(1)
        StSk = 3
    End If
    If z = 2 Then
        Toolbar2_ButtonClick Toolbar2.Buttons.Item(2)
        Line Input #1, ee
        If Not IsNumeric(ee) Or EOF(1) Then
            NuskaitytiPradiniusDuomenisIsFailo = False
            Close #1
            Exit Function
        End If
        Sk = Val(ee)
        Text2.Text = Sk
        StSk = 2
    End If
    If z = 3 Then
        Toolbar2_ButtonClick Toolbar2.Buttons.Item(3)
    End If
End Function

```

```

Line Input #1, ee
If Not IsNumeric(ee) Or EOF(1) Then
    NuskaitytiPradiniusDuomenisIsFailo = False
    Close #1
    Exit Function
End If
Sk = Val(ee)
Text2.Text = Sk
Line Input #1, ee
If Not IsNumeric(ee) Or EOF(1) Then
    NuskaitytiPradiniusDuomenisIsFailo = False
    Close #1
    Exit Function
End If
Sk = Val(ee)
Text3.Text = Sk
StSk = 3
End If
' Sluoksniu nuskaitymas
Line Input #1, ee
If Not IsNumeric(ee) Or EOF(1) Then
    NuskaitytiPradiniusDuomenisIsFailo = False
    Close #1
    Exit Function
End If
z = Val(ee)
For l = 1 To z
    MSHFlexGrid2.Row = 1
    MSHFlexGrid2.Col = 0
    MSHFlexGrid2.Text = l
    For j = 1 To StSk
        MSHFlexGrid2.Col = j
        If j = StSk Then
            Line Input #1, ee
            If Not IsNumeric(ee) Or EOF(1) Then
                NuskaitytiPradiniusDuomenisIsFailo = False
                Close #1
                Exit Function
            End If
            Skaic = Val(ee)
            If Skaic > MedziaguSkaicius Then
                NuskaitytiPradiniusDuomenisIsFailo = False
                Close #1
                Exit Function
            End If
            MSHFlexGrid1.Row = Skaic
            MSHFlexGrid1.Col = 1
            MSHFlexGrid2.Text = MSHFlexGrid1.Text
        Else
            Line Input #1, ee
            If Not IsNumeric(ee) Or EOF(1) Then
                NuskaitytiPradiniusDuomenisIsFailo = False
                Close #1
                Exit Function
            End If
            MSHFlexGrid2.Text = ee
        End If
    Next
    For k = 0 To StSk
        MSHFlexGrid2.Col = k
        MSHFlexGrid2.CellAlignment = flexAlignCenterCenter
    Next
    MSHFlexGrid2.Rows = MSHFlexGrid2.Rows + 1
Next
' Apkrovos nuskaitymas
Line Input #1, ee
If Not IsNumeric(ee) Or EOF(1) Then
    NuskaitytiPradiniusDuomenisIsFailo = False
    Close #1
    Exit Function
End If
Sk = Val(ee)
If Sk = 1 Then
    Option1.Value = True
Else
    Option2.Value = True
End If
Line Input #1, ee
If Not IsNumeric(ee) Or EOF(1) Then

```



```

        NuskaitytiPradiniusDuomenisIsFailo = False
        Close #1
        Exit Function
    End If
    Text6.Text = Val(ee)
    Line Input #1, ee
    If Not IsNumeric(ee) Then
        NuskaitytiPradiniusDuomenisIsFailo = False
        Close #1
        Exit Function
    End If
    Text7.Text = Val(ee)
    Close #1
    SluoksniuNumeris MSHFlexGrid2.Rows - 1
    IsrinktiEilute MSHFlexGrid2, MSHFlexGrid2.Rows - 1
    MedziagosNumeris MSHFlexGrid1.Rows - 1
    IsrinktiEilute MSHFlexGrid1, MSHFlexGrid1.Rows - 1
End Function

' Grazina True, jei pradiniai duomenys geri, priezingu atveju - False.
Public Function PradiniaiDuomenys() As Boolean
    PradiniaiDuomenys = True
    Ssk = MSHFlexGrid2.Rows - 2
    If Ssk < 1 Then
        x = MsgBox("Nėra nei vieno sluoksni!" & Chr(13) & "Įveskite bent vieną sluoksni.", vbExclamation + vbOKOnly, "Pranešimas")
        Text4.SetFocus
        PradiniaiDuomenys = False
        Exit Function
    End If
    Ssk = Ssk - 1
    ReDim Rv(Ssk)
    ReDim Ri(Ssk)
    ReDim bv(Ssk)
    ReDim bi(Ssk)
    ReDim hv(Ssk)
    ReDim hi(Ssk)
    ReDim A(Ssk)
    ReDim E(Ssk)
    ReDim s(Ssk)
    kurism = MSHFlexGrid1.RowSel
    kuriss = MSHFlexGrid2.RowSel
    For I = 0 To Ssk
        MSHFlexGrid2.Row = I + 1
        If Toolbar2.Buttons.Item(1).Value = tbrPressed Then
            MSHFlexGrid2.Col = 1
            bi(I) = Val(MSHFlexGrid2.Text)
            MSHFlexGrid2.Col = 2
            hi(I) = Val(MSHFlexGrid2.Text)
            MSHFlexGrid2.Col = 3
        End If
        If Toolbar2.Buttons.Item(2).Value = tbrPressed Then
            If I = 0 Then
                If Not TestText(Text2, "<", "Vidinis spindulys R[0] negali būti mažesnis už 0!") Then
                    PradiniaiDuomenys = False
                    IsrinktiEilute MSHFlexGrid1, kurism
                    IsrinktiEilute MSHFlexGrid2, kuriss
                    Text2.SetFocus
                    Exit Function
                End If
                Rv(I) = Val(Text2.Text)
            Else
                Rv(I) = Ri(I - 1)
            End If
            MSHFlexGrid2.Col = 1
            Ri(I) = Rv(I) + Val(MSHFlexGrid2.Text)
            MSHFlexGrid2.Col = 2
        End If
        If Toolbar2.Buttons.Item(3).Value = tbrPressed Then
            If I = 0 Then
                If Not TestText(Text2, "<", "Vidinis spyryo plotis b[0] negali būti mažesnis už 0!") Then
                    PradiniaiDuomenys = False
                    IsrinktiEilute MSHFlexGrid1, kurism
                    IsrinktiEilute MSHFlexGrid2, kuriss
                    Text2.SetFocus
                    Exit Function
                End If
                bv(I) = Val(Text2.Text)
                If Not TestText(Text3, "<", "Vidinis spyryo aukštis h[0] negali būti mažesnis už 0!") Then
                    PradiniaiDuomenys = False

```

```

        IsrinktiEilute MSHFlexGrid1, kurism
        IsrinktiEilute MSHFlexGrid2, kuriss
        Text3.SetFocus
        Exit Function
    End If
    hv(I) = Val(Text3.Text)
Else
    bv(I) = bi(I - 1)
    hv(I) = hi(I - 1)
End If
MSHFlexGrid2.Col = 1
bi(I) = bv(I) + Val(MSHFlexGrid2.Text)
MSHFlexGrid2.Col = 2
hi(I) = hv(I) + Val(MSHFlexGrid2.Text)
MSHFlexGrid2.Col = 3
End If
E(I) = TamprumoModulis(MSHFlexGrid2.Text)
Next
pr = ""
If Option1.Value = True Then
    pr = "Tempimo jėga F turi būti didesnė už 0!"
Else
    pr = "Strypo deformacija e turi būti didesnė už 0!"
End If
If Not TestText(Text6, "<=", pr) Then
    PradiniaiDuomenys = False
    IsrinktiEilute MSHFlexGrid1, kurism
    IsrinktiEilute MSHFlexGrid2, kuriss
    Text6.SetFocus
    Exit Function
End If
If Option1.Value = True Then
    N = Val(Text6.Text)
    eps = 0
Else
    N = 0
    eps = Val(Text6.Text)
End If
If Not TestText(Text7, "<=", "Strypo ilgis L turi būti didesnis už 0!") Then
    PradiniaiDuomenys = False
    IsrinktiEilute MSHFlexGrid1, kurism
    IsrinktiEilute MSHFlexGrid2, kuriss
    Text7.SetFocus
    Exit Function
End If
Ls = Val(Text7.Text)
IsrinktiEilute MSHFlexGrid1, kurism
IsrinktiEilute MSHFlexGrid2, kuriss
End Function

Public Sub PagrindiniaiParametrai()
    Const PI As Double = 3.14159265358979
    Ak = 0
    Bk = 0
    For I = 0 To Ssk
        If Toolbar2.Buttons.Item(1).Value = tbrPressed Then
            A(I) = bi(I) * hi(I)
        End If
        If Toolbar2.Buttons.Item(2).Value = tbrPressed Then
            A(I) = PI * (Ri(I) ^ 2 - Rv(I) ^ 2)
        End If
        If Toolbar2.Buttons.Item(3).Value = tbrPressed Then
            A(I) = bi(I) * hi(I) - bv(I) * hv(I)
        End If
        Ak = Ak + A(I)
        Bk = Bk + E(I) * A(I)
    Next I
    Ek = Bk / Ak
    If Option1.Value = True Then
        Sk = N / Ak
        eps = N / Ek / Ak
    Else
        Sk = Ek * eps
        N = Sk * Ak
    End If
    MaxS = 0
    For I = 0 To Ssk
        s(I) = Sk * E(I) / Ek
        If Abs(MaxS) < Abs(s(I)) Then

```

```

        MaxS = s(I)
        MaxSsluoksnis = I + 1
    End If
Next I
End Sub
Public Sub SkaiciavimoRezultatai()
    Dim endl As String, Text As String
    endl = Chr(13) & Chr(10)
    Text = ""
    RichTextBox1.SelBold = False
    RichTextBox1.SelUnderline = False
    ' Pradiniai duomenys
    Text = "PRADINIAI DUOMENYS:"
    Text = Text & endl & endl
    For I = 0 To Ssk
        kk = CStr(I + 1)
        If Toolbar2.Buttons.Item(1).Value = tbrPressed Then
            Text = Text & kk & "-o sluoksnio plotis b[" & kk & "] = " & Format(bi(I), "0.000") & " [mm]" & endl
            Text = Text & kk & "-o sluoksnio aukštis h[" & kk & "] = " & Format(hi(I), "0.000") & " [mm]" & endl
        End If
        If Toolbar2.Buttons.Item(2).Value = tbrPressed Then
            Text = Text & kk & "-o sluoksnio vidinis spindulys Rv[" & kk & "] = " & Format(Rv(I), "0.000") & " [mm]" & endl
            Text = Text & kk & "-o sluoksnio išorinis spindulys Ri[" & kk & "] = " & Format(Ri(I), "0.000") & " [mm]" & endl
        End If
        If Toolbar2.Buttons.Item(3).Value = tbrPressed Then
            Text = Text & kk & "-o sluoksnio vidinis plotis bv[" & kk & "] = " & Format(bv(I), "0.000") & " [mm]" & endl
            Text = Text & kk & "-o sluoksnio vidinis aukštis hv[" & kk & "] = " & Format(hv(I), "0.000") & " [mm]" & endl
            Text = Text & kk & "-o sluoksnio išorinis plotis bi[" & kk & "] = " & Format(bi(I), "0.000") & " [mm]" & endl
            Text = Text & kk & "-o sluoksnio išorinis aukštis hi[" & kk & "] = " & Format(hi(I), "0.000") & " [mm]" & endl
        End If
        Text = Text & kk & "-o sluoksnio skerspjūvio plotas A[" & kk & "] = " & Format(A(I), "0.000") & " [mm^2]" & endl
        Text = Text & kk & "-o sluoksnio medžiagos tamprumo modulis E[" & kk & "] = " & Format(E(I), "0.000") & " [MPa]" & endl & endl
    Next
    If Option1.Value = True Then
        Text = Text & "Tempimo jėga F = " & Format(N, "0.000") & " [N]" & endl & endl
    Else
        Text = Text & "Santykinė deformacija e = " & Format(eps, "0.00000") & endl & endl
    End If
    Text = Text & "SKAIČIAVIMO REZULTATAI:"
    Text = Text & endl & endl
    Text = Text & "Konstrukcijos skerspjūvio plotas Ak = " & Format(Ak, "0.000") & " [mm^2]" & endl
    Text = Text & "Konstrukcijos tamprumo modulis Ek = " & Format(Ek, "0.000") & " [MPa]" & endl
    If Option1.Value = False Then Text = Text & "Tempimo jėga F = " & Format(N, "0.000") & " [N]" & endl
    Text = Text & endl & "Normaliniai įtempimai S [MPa]:" & endl
    Smax = 0#
    SmaxSluoksnis = "0"
    ' Normaliniu itempimu skaiciavimas ir isvedimas
    For I = 0 To Ssk
        kk = CStr(I + 1)
        Text = Text & "S[" & kk & "] = " & Format(s(I), "0.000") & endl
    Next
    Text = Text & endl
    ' Deformaciju ir pailgejimo skaiciavimas
    If Option1.Value = True Then Text = Text & "Santykinė deformacija e = " & Format(eps, "0.00000") & "." & endl
    Text = Text & "Strypo pailgejimas dL = " & Format(eps * Ls, "0.00000") & " [mm]" & endl & endl
    ' Maksimaliu reiksmiu isvedimas
    Text = Text & "Maksimalus normalinis įtempimas Smax = " & Format(MaxS, "0.000") & " [MPa] yra " & CStr(MaxSsluoksnis) & "
    sluksnyje."
    RichTextBox1.Text = Text
End Sub

Public Sub DuomenysGrafikuBraizymui()
    Const TskSk As Integer = 50
    Const SpalvuSk As Integer = 5
    Dim GrSk As Integer
    ' Standartiniu spalvu aprasymas
    Dim Psp(SpalvuSk) As Long, Isp(SpalvuSk) As Long
    Psp(0) = RGB(0, 0, 128): Isp(0) = RGB(0, 0, 255)
    Psp(1) = RGB(0, 128, 0): Isp(1) = RGB(0, 200, 0)
    Psp(2) = RGB(128, 0, 0): Isp(2) = RGB(255, 0, 0)
    Psp(3) = RGB(0, 128, 128): Isp(3) = RGB(0, 200, 200)
    Psp(4) = RGB(128, 0, 128): Isp(4) = RGB(255, 0, 255)
    Psp(5) = RGB(32, 128, 32): Isp(5) = RGB(64, 255, 64)
    ' Maksimalaus plocio apskaiciavimas
    Dim MinA As Double, MaxA As Double, d As Double, N As Integer
    MinA = 0
    MaxA = Apvalinti(5# * Ak, GrSk)
    N = 5
    MinMaxRibos MinA, MaxA, N, d

```

```

dA = MaxA / (TskSk - 1)
GrSk = 2 * (Ssk + 1) - 1
ReDim gR(GrSk)
ReDim gTsk(GrSk)
ReDim gA(GrSk, TskSk)
ReDim gEk(GrSk, TskSk)
ReDim gBk(GrSk, TskSk)
ReDim gSp(GrSk)
ReDim gSt(GrSk)
ReDim gTip(GrSk)
ReDim gPavEk(GrSk)
ReDim gPavBk(GrSk)
j = 0
For I = 0 To Ssk
    gR(I) = True
    gSp(I) = Psp(j)
    gSt(I) = 2
    gTip(I) = vbDot
    gTsk(I) = 2
    gA(I, 0) = 0: gEk(I, 0) = E(I)
    gA(I, 1) = MaxA: gEk(I, 1) = E(I)
    gPavEk(I) = "E[" + CStr(I + 1) + "]"
    If j = SpalvuSk Then j = 0 Else j = j + 1
Next I
j = 0
For I = 0 To Ssk
    snr = Ssk + 1 + I
    gR(snr) = True
    gSp(snr) = Isp(j)
    gSt(snr) = 2
    gTip(snr) = vbSolid
    gTsk(snr) = TskSk
    gPavEk(snr) = "Ek( A[" + CStr(I + 1) + "]" )"
    gPavBk(snr) = "Bk( A[" + CStr(I + 1) + "]" )"
    Ai = 0#
    For k = 0 To TskSk
        gA(snr, k) = Ai
        Ak1 = Ai
        Bk1 = Ai * E(I)
        For z = 0 To Ssk
            If I <> z Then
                Ak1 = Ak1 + A(z)
                Bk1 = Bk1 + E(z) * A(z)
            End If
        Next
        gBk(snr, k) = Bk1 / 1000
        If Ak1 = 0 Then
            If Bk1 = 0 Then
                Ek1 = 0
            Else
                Ek1 = Bk1 / (dA / 10)
            End If
        Else
            Ek1 = Bk1 / Ak1
        End If
        gEk(snr, k) = Ek1
        Ai = Ai + dA
    Next k
    If j = SpalvuSk Then j = 0 Else j = j + 1
Next I
End Sub

' Skerspjuvio braizymas
Public Sub Skerspjuvis(Pav As PictureBox, Optional dx As Integer = 20, _
    Optional Itemp As Boolean = False)
    kuris = MSHFlexGrid2.RowSel
    Dim SluoksniuSk As Integer, X0 As Integer, Y0 As Integer, kiek As Integer
    Dim Plotis As Integer, Aukstis As Integer, X0l As Integer, X0sk As Integer
    Dim MaxSkIlgis As Integer, Daugiklis As Integer
    MaxSkIlgis = 0
    Pav.FillStyle = vbFSSolid
    Pav.FillColor = Pav.BackColor
    Pav.ForeColor = Pav.BackColor
    Pav.Line (0, 0)-(Pav.ScaleWidth, Pav.ScaleHeight), Pav.BackColor, B
    SluoksniuSk = MSHFlexGrid2.Rows - 3
    If SluoksniuSk < 0 Then Exit Sub
    Dim MaxX As Double, MaxY As Double, Mastelis As Double, MastelisI As Double
    Dim b() As Double, h() As Double, IT() As Double, B0 As Double, H0 As Double
    ReDim b(SluoksniuSk)

```

```

ReDim h(SluoksnuiSk)
MaxX = 0#
MaxY = 0#
Pav.DrawWidth = 2
Pav.ForeColor = 0
Pav.FillColor = 0
Pav.FillStyle = vbUpwardDiagonal
If Itemp Then
    Plotis = Pav.ScaleWidth / 2 - 2 * dx
    X0I = Pav.ScaleWidth - dx
    Y0I = Pav.ScaleHeight - dx
Else
    Plotis = Pav.ScaleWidth - 2 * dx
End If
Aukstis = Pav.ScaleHeight - 2 * dx
' Pirmas skerspjūvis
If Toolbar2.Buttons.Item(1).Value = tbrPressed Then
    kiek = 2 * (SluoksnuiSk + 1) + 1
    If Itemp Then ReDim IT(kiek)
    For I = 0 To SluoksnuiSk
        MSHFlexGrid2.Row = I + 1
        MSHFlexGrid2.Col = 1
        b(I) = Val(MSHFlexGrid2.Text)
        MSHFlexGrid2.Col = 2
        h(I) = Val(MSHFlexGrid2.Text)
        If MaxX < b(I) Then MaxX = b(I)
        MaxY = MaxY + h(I)
    Next I
    If MaxX = 0 Then mX = 1 Else mX = Plotis / MaxX
    If MaxY = 0 Then mY = 1 Else mY = Aukstis / MaxY
    If mX > mY Then Mastelis = mY Else Mastelis = mX
    If Itemp Then X0 = Pav.ScaleWidth / 4 Else X0 = Pav.ScaleWidth / 2
    Y0 = dx
    For I = 0 To SluoksnuiSk
        xx = Mastelis * b(I) / 2#
        If Pav.FillStyle = vbUpwardDiagonal Then
            Pav.FillStyle = vbDownwardDiagonal
        Else
            Pav.FillStyle = vbUpwardDiagonal
        End If
        yy = Mastelis * h(I)
        If Itemp Then
            Pav.DrawWidth = 1
            Pav.DrawStyle = vbDot
            Pav.Line (X0, Y0)-(X0I, Y0)
            If I = SluoksnuiSk Then Pav.Line (X0, Y0 + yy)-(X0I, Y0 + yy)
            Pav.DrawWidth = 2
            Pav.DrawStyle = vbSolid
            skilgis = Pav.TextWidth(Format(s(I), "0.000 MPa"))
            If skilgis > MaxSk Then MaxSk = skilgis
        End If
        Pav.Line (X0 - xx, Y0)-(X0 + xx, Y0 + yy), B
        Y0 = Y0 + yy
    Next I
    If Itemp Then
        MastelisI = (Plotis - MaxSk - dx / 2) / MaxS
        Y0 = dx
        If N > 0 Then
            X0I = Pav.ScaleWidth / 2 + dx
            X0sk = Pav.ScaleWidth - dx
            Daugiklis = -1
        Else
            X0I = Pav.ScaleWidth - dx
            X0sk = Pav.ScaleWidth / 2 + dx
            Daugiklis = 0
        End If
        Pav.DrawWidth = 3
        Pav.DrawStyle = vbSolid
        xx = X0I
        For I = 0 To SluoksnuiSk
            yy = Mastelis * h(I)
            xx1 = s(I) * MastelisI
            Pav.Line (xx, Y0)-(X0I + xx1, Y0)
            Pav.Line (X0I + xx1, Y0)-(X0I + xx1, Y0 + yy)
            sks = Format(s(I), "0.000 MPa")
            skilgis = Pav.TextWidth(sks)
            skaukstis = Pav.TextHeight(sks)
            Pav.CurrentX = X0sk + Daugiklis * skilgis
            Pav.CurrentY = Y0 + (yy - skaukstis) / 2
        Next I
    End If

```

```

        Pav.Print sks
        xx = X0I + xx1
        Y0 = Y0 + yy
        If I = SluoksniauSk Then Pav.Line (xx, Y0)-(X0I, Y0)
    Next I
    Pav.DrawWidth = 1
    Pav.Line (X0I, Y0)-(X0I, dx)
End If
End If
' Antras skerspjūvis
If Toolbar2.Buttons.Item(2).Value = tbrPressed Then
    B0 = Val(Text2.Text)
    If B0 < 0 Then B0 = 0
    MaxX = B0
    For I = 0 To SluoksniauSk
        MSHFlexGrid2.Row = I + 1
        MSHFlexGrid2.Col = 1
        b(I) = Val(MSHFlexGrid2.Text)
        MaxX = MaxX + b(I)
        b(I) = MaxX
    Next I
    mX = Plotis / MaxX / 2#
    mY = Aukstis / MaxX / 2#
    If mX > mY Then Mastelis = mY Else Mastelis = mX
    X0 = Plotis / 2# + dx
    Y0 = Aukstis / 2# + dx
    For I = SluoksniauSk To 0 Step -1
        If Pav.FillStyle = vbUpwardDiagonal Then
            Pav.FillStyle = vbDownwardDiagonal
        Else
            Pav.FillStyle = vbUpwardDiagonal
        End If
        Rii = Mastelis * b(I)
        Pav.Circle (X0, Y0), Rii
        If I > 0 Then
            St = Pav.FillStyle
            Sp = Pav.FillColor
            Pav.FillStyle = vbFSSolid
            Pav.FillColor = Pav.BackColor
            Pav.Circle (X0, Y0), Mastelis * b(I - 1)
            Pav.FillStyle = St
            Pav.FillColor = Sp
        End If
        If Itemp Then
            Pav.DrawWidth = 1
            Pav.DrawStyle = vbDot
            Pav.Line (X0, Y0 + Rii)-(X0I, Y0 + Rii)
            Pav.Line (X0, Y0 - Rii)-(X0I, Y0 - Rii)
            Pav.DrawWidth = 2
            Pav.DrawStyle = vbSolid
            skilgis = Pav.TextWidth(Format(s(I), "0.000 MPa"))
            If skilgis > MaxSk Then MaxSk = skilgis
        End If
    Next I
    If B0 > 0 Then
        Rii = Mastelis * B0
        Pav.FillStyle = vbFSSolid
        Pav.FillColor = Pav.BackColor
        Pav.Circle (X0, Y0), Rii
        If Itemp Then
            Pav.DrawWidth = 1
            Pav.DrawStyle = vbDot
            Pav.Line (X0, Y0 + Rii)-(X0I, Y0 + Rii)
            Pav.Line (X0, Y0 - Rii)-(X0I, Y0 - Rii)
            Pav.DrawWidth = 2
            Pav.DrawStyle = vbSolid
        End If
    End If
    If Itemp Then
        MastelisI = (Plotis - MaxSk - dx / 2) / MaxS
        If N > 0 Then
            X0I = Pav.ScaleWidth / 2 + dx
            X0sk = Pav.ScaleWidth - dx
            Daugiklis = -1
        Else
            X0I = Pav.ScaleWidth - dx
            X0sk = Pav.ScaleWidth / 2 + dx
            Daugiklis = 0
        End If
    End If
End If

```

```

Pav.DrawWidth = 3
Pav.DrawStyle = vbSolid
xx = X0I
For I = SluoksniuSk To 1 Step -1
    yy = Mastelis * b(I)
    xx1 = s(I) * MastelisI
    Pav.Line (xx, Y0 - yy)-(X0I + xx1, Y0 - yy)
    Pav.Line (xx, Y0 + yy)-(X0I + xx1, Y0 + yy)
    xx = X0I + xx1
    yy1 = Mastelis * b(I - 1)
    Pav.Line (xx, Y0 - yy)-(xx, Y0 - yy1)
    Pav.Line (xx, Y0 + yy)-(xx, Y0 + yy1)
    sks = Format(s(I), "0.000 MPa")
    skilgis = Pav.TextWidth(sks)
    skaukstis = Pav.TextHeight(sks)
    Pav.CurrentX = X0sk + Daugiklis * skilgis
    Pav.CurrentY = Y0 + (yy + yy1 - skaukstis) / 2
    Pav.Print sks
    Pav.CurrentX = X0sk + Daugiklis * skilgis
    Pav.CurrentY = Y0 - (yy + yy1 + skaukstis) / 2
    Pav.Print sks
    If I = 1 Then
        yy = Mastelis * b(0)
        xx1 = s(0) * MastelisI
        Pav.Line (xx, Y0 - yy)-(X0I + xx1, Y0 - yy)
        Pav.Line (xx, Y0 + yy)-(X0I + xx1, Y0 + yy)
        xx = X0I + xx1
        yy1 = Mastelis * B0
        Pav.Line (xx, Y0 - yy)-(xx, Y0 - yy1)
        Pav.Line (xx, Y0 + yy)-(xx, Y0 + yy1)
        sks = Format(s(0), "0.000 MPa")
        skilgis = Pav.TextWidth(sks)
        skaukstis = Pav.TextHeight(sks)
        Pav.CurrentX = X0sk + Daugiklis * skilgis
        Pav.CurrentY = Y0 + (yy + yy1 - skaukstis) / 2
        Pav.Print sks
        Pav.CurrentX = X0sk + Daugiklis * skilgis
        Pav.CurrentY = Y0 - (yy + yy1 + skaukstis) / 2
        Pav.Print sks
    End If
Next I
If SluoksniuSk = 0 Then
    yy = Mastelis * b(I)
    xx1 = s(I) * MastelisI
    Pav.Line (xx, Y0 - yy)-(X0I + xx1, Y0 - yy)
    Pav.Line (xx, Y0 + yy)-(X0I + xx1, Y0 + yy)
    xx = X0I + xx1
    yy1 = Mastelis * B0
    Pav.Line (xx, Y0 - yy)-(xx, Y0 - yy1)
    Pav.Line (xx, Y0 + yy)-(xx, Y0 + yy1)
    sks = Format(s(I), "0.000 MPa")
    skilgis = Pav.TextWidth(sks)
    skaukstis = Pav.TextHeight(sks)
    Pav.CurrentX = X0sk + Daugiklis * skilgis
    Pav.CurrentY = Y0 + (yy + yy1 - skaukstis) / 2
    Pav.Print sks
    Pav.CurrentX = X0sk + Daugiklis * skilgis
    Pav.CurrentY = Y0 - (yy + yy1 + skaukstis) / 2
    Pav.Print sks
End If
If B0 > 0 Then
    Pav.Line (xx, Y0 - yy1)-(X0I, Y0 - yy1)
    Pav.Line (xx, Y0 + yy1)-(X0I, Y0 + yy1)
    Pav.Line (X0I, Y0 + yy1)-(X0I, Y0 - yy1)
End If
Pav.DrawWidth = 1
yy = Mastelis * MaxX
Pav.Line (X0I, Y0 + yy)-(X0I, Y0 - yy)
End If
End If
' Trecias skerspjuvis
If Toolbar2.Buttons.Item(3).Value = tbrPressed Then
    B0 = Val(Text2.Text)
    H0 = Val(Text3.Text)
    If B0 <= 0 Or H0 <= 0 Then
        B0 = 0
        H0 = 0
    End If
    MaxX = B0

```

```

MaxY = H0
For I = 0 To SluoksnuiSk
  MSHFlexGrid2.Row = I + 1
  MSHFlexGrid2.Col = 1
  b(I) = Val(MSHFlexGrid2.Text)
  MSHFlexGrid2.Col = 2
  h(I) = Val(MSHFlexGrid2.Text)
  MaxX = MaxX + b(I)
  b(I) = MaxX
  MaxY = MaxY + h(I)
  h(I) = MaxY
Next I
mX = Plotis / MaxX
mY = Aukstis / MaxY
If mX > mY Then Mastelis = mY Else Mastelis = mX
X0 = Plotis / 2# + dx
Y0 = Aukstis / 2# + dx
For I = SluoksnuiSk To 0 Step -1
  If Pav.FillStyle = vbUpwardDiagonal Then
    Pav.FillStyle = vbDownwardDiagonal
  Else
    Pav.FillStyle = vbUpwardDiagonal
  End If
  xx = Mastelis * b(I) / 2#
  yy = Mastelis * h(I) / 2#
  Pav.Line (X0 - xx, Y0 + yy)-(X0 + xx, Y0 - yy), , B
  If Itemp Then
    Pav.DrawWidth = 1
    Pav.DrawStyle = vbDot
    Pav.Line (X0, Y0 + yy)-(X0I, Y0 + yy)
    Pav.Line (X0, Y0 - yy)-(X0I, Y0 - yy)
    Pav.DrawWidth = 2
    Pav.DrawStyle = vbSolid
    skilgis = Pav.TextWidth(Format(s(I), "0.000 MPa"))
    If skilgis > MaxSk Then MaxSk = skilgis
  End If
  If I > 0 Then
    St = Pav.FillStyle
    Sp = Pav.FillColor
    Pav.FillStyle = vbFSSolid
    Pav.FillColor = Pav.BackColor
    xx = Mastelis * b(I - 1) / 2#
    yy = Mastelis * h(I - 1) / 2#
    Pav.Line (X0 - xx, Y0 + yy)-(X0 + xx, Y0 - yy), , B
    Pav.FillStyle = St
    Pav.FillColor = Sp
    If Itemp Then
      Pav.DrawWidth = 1
      Pav.DrawStyle = vbDot
      Pav.Line (X0, Y0 + yy)-(X0I, Y0 + yy)
      Pav.Line (X0, Y0 - yy)-(X0I, Y0 - yy)
      Pav.DrawWidth = 2
      Pav.DrawStyle = vbSolid
    End If
  End If
Next I
If B0 > 0 And H0 > 0 Then
  Pav.FillStyle = vbFSSolid
  Pav.FillColor = Pav.BackColor
  xx = Mastelis * B0 / 2#
  yy = Mastelis * H0 / 2#
  Pav.Line (X0 - xx, Y0 + yy)-(X0 + xx, Y0 - yy), , B
  If Itemp Then
    Pav.DrawWidth = 1
    Pav.DrawStyle = vbDot
    Pav.Line (X0, Y0 + yy)-(X0I, Y0 + yy)
    Pav.Line (X0, Y0 - yy)-(X0I, Y0 - yy)
    Pav.DrawWidth = 2
    Pav.DrawStyle = vbSolid
  End If
End If
If Itemp Then
  MastelisI = (Plotis - MaxSk - dx / 2) / MaxS
  If N > 0 Then
    X0I = Pav.ScaleWidth / 2 + dx
    X0sk = Pav.ScaleWidth - dx
    Daugiklis = -1
  Else
    X0I = Pav.ScaleWidth - dx

```



```

X0sk = Pav.ScaleWidth / 2 + dx
Daugiklis = 0
End If
Pav.DrawWidth = 3
Pav.DrawStyle = vbSolid
xx = X0I
For I = SluoksniuSk To 1 Step -1
    yy = Mastelis * h(I) / 2#
    xx1 = s(I) * MastelisI
    Pav.Line (xx, Y0 - yy)-(X0I + xx1, Y0 - yy)
    Pav.Line (xx, Y0 + yy)-(X0I + xx1, Y0 + yy)
    xx = X0I + xx1
    yy1 = Mastelis * h(I - 1) / 2#
    Pav.Line (xx, Y0 - yy)-(xx, Y0 - yy1)
    Pav.Line (xx, Y0 + yy)-(xx, Y0 + yy1)
    sks = Format(s(I), "0.000 MPa")
    skilgis = Pav.TextWidth(sks)
    skaukstis = Pav.TextHeight(sks)
    Pav.CurrentX = X0sk + Daugiklis * skilgis
    Pav.CurrentY = Y0 + (yy + yy1 - skaukstis) / 2
    Pav.Print sks
    Pav.CurrentX = X0sk + Daugiklis * skilgis
    Pav.CurrentY = Y0 - (yy + yy1 + skaukstis) / 2
    Pav.Print sks
    If I = 1 Then
        yy = Mastelis * h(0) / 2#
        xx1 = s(0) * MastelisI
        Pav.Line (xx, Y0 - yy)-(X0I + xx1, Y0 - yy)
        Pav.Line (xx, Y0 + yy)-(X0I + xx1, Y0 + yy)
        xx = X0I + xx1
        yy1 = Mastelis * H0 / 2#
        Pav.Line (xx, Y0 - yy)-(xx, Y0 - yy1)
        Pav.Line (xx, Y0 + yy)-(xx, Y0 + yy1)
        sks = Format(s(0), "0.000 MPa")
        skilgis = Pav.TextWidth(sks)
        skaukstis = Pav.TextHeight(sks)
        Pav.CurrentX = X0sk + Daugiklis * skilgis
        Pav.CurrentY = Y0 + (yy + yy1 - skaukstis) / 2
        Pav.Print sks
        Pav.CurrentX = X0sk + Daugiklis * skilgis
        Pav.CurrentY = Y0 - (yy + yy1 + skaukstis) / 2
        Pav.Print sks
    End If
Next I
If SluoksniuSk = 0 Then
    yy = Mastelis * h(I) / 2#
    xx1 = s(I) * MastelisI
    Pav.Line (xx, Y0 - yy)-(X0I + xx1, Y0 - yy)
    Pav.Line (xx, Y0 + yy)-(X0I + xx1, Y0 + yy)
    xx = X0I + xx1
    yy1 = Mastelis * H0 / 2#
    Pav.Line (xx, Y0 - yy)-(xx, Y0 - yy1)
    Pav.Line (xx, Y0 + yy)-(xx, Y0 + yy1)
    sks = Format(s(I), "0.000 MPa")
    skilgis = Pav.TextWidth(sks)
    skaukstis = Pav.TextHeight(sks)
    Pav.CurrentX = X0sk + Daugiklis * skilgis
    Pav.CurrentY = Y0 + (yy + yy1 - skaukstis) / 2
    Pav.Print sks
    Pav.CurrentX = X0sk + Daugiklis * skilgis
    Pav.CurrentY = Y0 - (yy + yy1 + skaukstis) / 2
    Pav.Print sks
End If
If B0 > 0 Then
    Pav.Line (xx, Y0 - yy1)-(X0I, Y0 - yy1)
    Pav.Line (xx, Y0 + yy1)-(X0I, Y0 + yy1)
    Pav.Line (X0I, Y0 + yy1)-(X0I, Y0 - yy1)
End If
Pav.DrawWidth = 1
yy = Mastelis * MaxX / 2#
Pav.Line (X0I, Y0 + yy)-(X0I, Y0 - yy)
End If
End If
IsrinktiEilute MSHFlexGrid2, kuris
End Sub

Public Sub GrafineDalis()
    Dim dd As Stac
    dd.x = 120

```

```

dd.y = 120
dd.dx = 0
dd.dy = 0
kiek = UBound(gR)
kiek1 = (kiek - 1) / 2
Select Case Combo2.ListIndex
Case 0
    Skerspjuvis Graf, 120, True
Case 1
    If Combo3.ListIndex = 0 Then
        For I = 0 To kiek
            gR(I) = True
        Next I
    Else
        For I = 0 To kiek
            gR(I) = False
        Next I
        gR(Combo3.ListIndex - 1) = True
        gR(kiek1 + Combo3.ListIndex) = True
    End If
    Grafikai Graf, dd, gTsk(), gR(), gA(), gEk(), gSp(), gSt(), gTip(), _
        gPavEk(), True, "A [mm^2]", "E [MPa]"
Case 2
    For I = 0 To kiek1
        gR(I) = False
    Next I
    If Combo3.ListIndex = 0 Then
        For I = kiek1 + 1 To kiek
            gR(I) = True
        Next I
    Else
        For I = kiek1 + 1 To kiek
            gR(I) = False
        Next I
        gR(kiek1 + Combo3.ListIndex) = True
    End If
    Grafikai Graf, dd, gTsk(), gR(), gA(), gBk(), gSp(), gSt(), gTip(), _
        gPavBk(), True, "A [mm^2]", "Bk [kN]"
End Select
End Sub

Public Sub Meniu_Mygtukai_TextBox()
    fMainForm.mnuFilePrint.Enabled = False
    fMainForm.mnuEditCut.Enabled = True
    fMainForm.mnuEditCopy.Enabled = True
    fMainForm.mnuEditPaste.Enabled = True
    fMainForm.tbToolBar.Buttons.Item(6).Enabled = False ' Spausdinti
    fMainForm.tbToolBar.Buttons.Item(8).Enabled = True ' Iskirpti
    fMainForm.tbToolBar.Buttons.Item(9).Enabled = True ' Kopijuoti
    fMainForm.tbToolBar.Buttons.Item(10).Enabled = True ' Istatyti
End Sub

Public Sub Meniu_Mygtukai_RichTextBox()
    fMainForm.mnuFilePrint.Enabled = True
    fMainForm.mnuEditCut.Enabled = False
    fMainForm.mnuEditCopy.Enabled = True
    fMainForm.mnuEditPaste.Enabled = False
    fMainForm.tbToolBar.Buttons.Item(6).Enabled = True ' Spausdinti
    fMainForm.tbToolBar.Buttons.Item(8).Enabled = False ' Iskirpti
    fMainForm.tbToolBar.Buttons.Item(9).Enabled = True ' Kopijuoti
    fMainForm.tbToolBar.Buttons.Item(10).Enabled = False ' Istatyti
End Sub

Public Sub Kopijuoti()
    Dim t
    For Each t In Controls
        If TypeOf t Is TextBox Then
            Clipboard.Clear
            Clipboard.SetText tSelText
        Exit For
        End If
        If TypeOf t Is RichTextBox Then
            Clipboard.Clear
            If RichTextBox1.SelLength <> 0 Then
                Clipboard.SetText RichTextBox1.SelText
            Else
                Clipboard.SetText RichTextBox1.Text
            End If
        Exit For
    End For

```

```

    End If
  Next t
End Sub

Public Function strdel(ByVal eile As String, ByVal nuo As Integer, ByVal kiek As Integer)
  strdel = Mid(eile, 1, nuo) & Mid(eile, nuo + kiek + 1)
End Function

Public Function insertstr(ByVal I_ka As String, ByVal kur As Integer, ByVal ka As String)
  If kur > Len(I_ka) Then
    insertstr = I_ka
  Else
    kaire = Left(I_ka, kur)
    desine = Right(I_ka, Len(I_ka) - kur)
    insertstr = kaire & ka & desine
  End If
End Function

Public Sub Istatyti()
  Dim t
  For Each t In Controls
    If TypeOf t Is TextBox Then
      If t.Tag = 1 Then
        kur = t.SelStart
        t.Text = strdel(t.Text, kur, Len(t.SelText))
        t.Text = insertstr(t.Text, kur, Clipboard.GetText)
        t.SelStart = kur + Len(Clipboard.GetText)
      Exit For
    End If
  End If
Next t
End Sub

Public Sub Iskirpti()
  Dim t
  For Each t In Controls
    If TypeOf t Is TextBox Then
      If t.Tag = 1 Then
        Clipboard.SetText t.SelText
        kur = t.SelStart
        t.Text = strdel(t.Text, kur, Len(t.SelText))
        t.SelStart = kur
      Exit For
    End If
  End If
Next t
End Sub

Public fMainForm As frmMain
Public frmApi As frmAbout

Public Type Stac
  x As Double
  y As Double
  dx As Double
  dy As Double
End Type

Sub Main()
  frmPradinis.Show
  Set fMainForm = New frmMain
  Set frmApi = New frmAbout
  Load fMainForm
End Sub

Public Function GStac(x As Double, y As Double, dx As Double, dy As Double) As Stac
  Dim s As Stac
  s.x = x
  s.y = y
  s.dx = dx
  s.dy = dy
  GStac = s
End Function

' gr - kur grafika atvaizduoti
' pavDydis - grafiko lango dydis
' Tsk(Sk) - tasku skaicius grafike
' Rodyti(Sk) - ar rodyti grafika

```

```

' x(Sk,Tsk) - grafiku x koordiantes
' y(Sk,Tsk) - grafiku y koordiantes
' Sp(Sk) - grafiku spalvos
' St(Sk) - grafiku liniju storiai
' Tip(Sk) - grafiku liniju tipai
' Pav(Sk) - kiekvieno grafiko zymejimas
' RodytiPav - nusako ar vaizduoti grafiku pavadinimus
' Xpav - x asies dydio zymejimas
' Ypav - x asies dydio zymejimas
Public Sub Grafikai(gR As PictureBox, pavDydis As Stac, _
    Tsk() As Integer, Rodyti() As Boolean, _
    x() As Double, y() As Double, Sp() As Long, _
    St() As Integer, Tip() As Integer, _
    Pav() As String, Optional RodytiPav As Boolean = True, _
    Optional Xpav As String = "", _
    Optional Ypav As String = "")
' Grafiku skaicius
Dim Sk, Sk1 As Integer
Sk = UBound(x, 1)
Sk1 = UBound(y, 1)
If Sk1 < Sk Then Sk = Sk1
' Grafiku sumazejimas del uzrasu (pixel'iais)
Dim skX, skX1, skY, skY1, zym As Integer
skX = 0: skX1 = 0: skY = 0: skY1 = 0: zym = 0
' Apskaiciuoju max ir min grafiko reiksmes
Dim Xmin As Double, Ymin As Double, _
    Xmax As Double, Ymax As Double
Xmin = x(0, 0): Ymin = y(0, 0): Xmax = x(0, 0): Ymax = y(0, 0)
For I = 0 To Sk
    If Rodyti(I) Then
        For j = 0 To Tsk(I) - 1
            If x(I, j) > Xmax Then
                Xmax = x(I, j)
            Else
                If x(I, j) < Xmin Then Xmin = x(I, j)
            End If
            If y(I, j) > Ymax Then
                Ymax = y(I, j)
            Else
                If y(I, j) < Ymin Then Ymin = y(I, j)
            End If
        Next j
    End If
Next I
' Perskaiciuoju min ir max reiksmes nes braizomas tinklelis
Dim Nx As Integer, Ny As Integer, _
    kiekX As Integer, kiekY As Integer
Dim dx As Double, dy As Double ' tinklelio zingsniai
Dim xaf As String, yaf As String
Nx = 5
Ny = 5
MinMaxRibos Xmin, Xmax, Nx, dx, kiekX
MinMaxRibos Ymin, Ymax, Ny, dy, kiekY
' Asiu skaiciu formatas
If kiekX = 0 Then
    xaf = "0"
Else
    xaf = "0."
    For I = 0 To kiekX - 1
        xaf = xaf + "0"
    Next I
End If
If kiekY = 0 Then
    yaf = "0"
Else
    yaf = "0."
    For I = 0 To kiekY - 1
        yaf = yaf + "0"
    Next I
End If
' Nustatau reikiama vietos dydi asiu zymejimui
Const ddy As Integer = 50 ' skaiciu atstumas nuo y asies
Const dxx As Integer = 50 ' skaiciu atstumas nuo x asies
Const ZymKreivIlgis As Integer = 300 ' Liniju ilgis zymejime
td = gR.TextWidth(Ypav)
If td > skX Then skX = td
For I = 0 To Ny
    yy = Ymin + I * dy
    td = gR.TextWidth(Format(yy, yaf))

```

```

    If td > skX Then skX = td
Next I
skX = skX + dyy
skY1 = gR.TextHeight(Format(yy, yaf)) / 2
td = gR.TextHeight(Xpav)
If td > skY Then skY = td
For I = 0 To Nx
    xx = Xmin + I * dx
    td = gR.TextHeight(Format(xx, xaf))
    If td > skY Then skY = td
Next I
skY = skY + dxx
skX1 = gR.TextWidth(Format(xx, xaf)) / 2
' Nustatau reikiama vietos dydi grafiku zymejimui atvaizduoti
If RodytiPav Then
    For I = 0 To Sk
        If Rodyti(I) Then
            td = gR.TextWidth(Pav(I)) + ZymKreivIlgis + dxx
            If td > zym Then zym = td
        End If
    Next I
End If
' Jei neuzduoti grafiko gabaritai, tai juos apskaiciuoju
If pavDydis.dx = 0 Then pavDydis.dx = gR.ScaleWidth - 2 * pavDydis.x
If pavDydis.dy = 0 Then pavDydis.dy = gR.ScaleHeight - 2 * pavDydis.y
' Ishalau grafiku braizymo erdve
gR.FillStyle = vbFSSolid
gR.FillColor = gR.BackColor
gR.ForeColor = gR.BackColor
gR.Line (-1, -1)-(gR.ScaleWidth + 1, gR.ScaleHeight + 1), , B
' Apskaiciuju x ir y asiu mastelius
Dim mX, mY As Double
mX = (pavDydis.dx - skX - skX1 - zym) / Abs(Xmax - Xmin)
mY = (pavDydis.dy - skY - skY1) / Abs(Ymax - Ymin)
' Apskaiciuju koordinaciju pradios taskus
Dim X0, Y0 As Long
X0 = pavDydis.x + skX - mX * Xmin
Y0 = pavDydis.y + skY1 + mY * Ymax
' Braizau tinkleli
gR.ForeColor = RGB(0, 0, 0)
gR.DrawStyle = vbSolid
gR.DrawWidth = 1
yy1 = Y0 - mY * Ymin 'pavDydis.y
yy2 = Y0 - mY * Ymax 'pavDydis.y + tdy * Ny - skY
For I = 0 To Nx
    xx = X0 + mX * (Xmin + I * dx)
    gR.Line (xx, yy1)-(xx, yy2)
Next I
xx1 = X0 + mX * Xmin 'pavDydis.x + skX
xx2 = X0 + mX * Xmax 'xx1 + tdx * Nx
For I = 0 To Ny
    yy = Y0 - mY * (Ymin + I * dy) 'pavDydis.y + i * tdy
    gR.Line (xx1, yy)-(xx2, yy)
Next I
' Braizau grafikus
Dim PixTiwp As Integer
yy1 = gR.CurrentY
gR.ScaleMode = vbPixels
gR.CurrentY = gR.CurrentY + 1
gR.ScaleMode = vbTwips
PixTiwp = Abs(gR.CurrentY - yy1)
For I = 0 To Sk
    If Rodyti(I) Then
        gR.DrawWidth = St(I)
        gR.DrawStyle = Tip(I)
        gR.ForeColor = Sp(I)
        For j = 0 To Tsk(I) - 2
            If Tip(I) <> vbSolid Then
                gR.DrawWidth = 1
                For k = 0 To St(I) - 1
                    gR.Line (X0 + mX * x(I, j), Y0 - mY * y(I, j) + k * PixTiwp)-
                        (X0 + mX * x(I, j + 1), Y0 - mY * y(I, j + 1) + k * PixTiwp)
                Next k
            Else
                gR.Line (X0 + mX * x(I, j), Y0 - mY * y(I, j))-
                    (X0 + mX * x(I, j + 1), Y0 - mY * y(I, j + 1))
            End If
        Next j
    End If
Next I
End If

```

```

Next I
' Isvadame skaičius ant asių
gR.ForeColor = RGB(0, 0, 0)
xx1 = X0 + mX * Xmin - dxx
For I = 0 To Ny
    yy = Ymin + I * dy
    yy1 = Y0 - mY * yy
    If I = Ny - 1 Then
        yys = Ypav
    Else
        yys = Format(yy, yaf)
    End If
    td1 = gR.TextWidth(yys)
    td2 = gR.TextHeight(yys)
    gR.CurrentX = xx1 - td1
    gR.CurrentY = yy1 - td2 / 2
    gR.Print yys
Next I
yy1 = Y0 - mY * Ymin + dyy
For I = 0 To Nx
    xx = Xmin + I * dx
    xx1 = X0 + mX * xx
    If I = Nx - 1 Then
        xxs = Xpav
    Else
        xxs = Format(xx, xaf)
    End If
    td1 = gR.TextWidth(xxs)
    gR.CurrentX = xx1 - td1 / 2
    gR.CurrentY = yy1
    gR.Print xxs
Next I
' Isvedame grafiku žymėjimą
xx1 = X0 + mX * Xmax + skX1 + dxx
xx3 = xx1 + ZymKreivlIlgis
xx2 = xx3 - dxx
yy1 = pavDydis.y
If RodytiPav Then
    For I = 0 To Sk
        If Rodyti(I) Then
            gR.DrawWidth = St(I)
            gR.DrawStyle = Tip(I)
            gR.ForeColor = Sp(I)
            yy3 = gR.TextHeight(Pav(I))
            yy2 = yy1 + yy3 / 2
            If Tip(I) <> vbSolid Then
                gR.DrawWidth = 1
                For k = 0 To St(I) - 1
                    gR.Line (xx1, yy2 + k * PixTiwp)-(xx2, yy2 + k * PixTiwp)
                Next k
            Else
                gR.Line (xx1, yy2 + k * PixTiwp)-(xx2, yy2 + k * PixTiwp)
            End If
            gR.CurrentX = xx3
            gR.CurrentY = yy1
            gR.ForeColor = 0
            gR.Print Pav(I)
            yy1 = yy1 + yy3 + dyy
        End If
    Next I
End If
End Sub

Public Function Apvalinti(ka As Double, kiek As Integer, Optional proc As Double = 10) As Double
    Dim xxx, k1 As Double
    Dim Sk As Integer
    xxx = ka * proc / 100#
    k1 = Round(1# / xxx)
    If k1 = 0 Then
        kiek = 0
        skp = 0
        While (ka / 10 ^ skp) > 10
            skp = skp + 1
        Wend
        Sk = Round(ka / 10 ^ skp)
        Apvalinti = Sk * 10 ^ skp
    Else
        kiek = 0
        While CInt(CInt(10 ^ kiek) / CInt(k1)) = 0

```

```

        kiek = kiek + 1
    Wend
    xxx = Round(ka, kiek)
    Apvalinti = xxx
End If
End Function

' Prskaiciuoja minimalia, maksimalia ribas, intervalo dydi ir intervalu
' skaiciu taip, kad sie dydziai butu 'grazus', t.y. su ribotu skaiciumi
' po kablelio.
Public Sub MinMaxRibos(MinSk As Double, MaxSk As Double, N As Integer, _
    dSk As Double, Optional kiek As Integer)
    If Abs(MaxSk - MinSk) <= 0.0000000000001 Then
        MinSk = MinSk - 0.5
        MaxSk = MaxSk + 0.5
    End If
    dSk = Apvalinti(Abs(MaxSk - MinSk) / N, (kiek))
    I = 1
    aMin = Abs(MinSk)
    While dSk * I < aMin
        I = I + 1
    Wend
    If MinSk > 0 Then MinSk = dSk * (I - 1)
    If MinSk < 0 Then MinSk = -dSk * I
    N = 1
    While MinSk + dSk * N < MaxSk
        N = N + 1
    Wend
    If 100 * Abs(MinSk + dSk * (N - 1) - MaxSk) / MaxSk < 0.1 Then N = N - 1
    MaxSk = MinSk + N * dSk
End Sub

```