

Izrada programa za problem gredni nosač iz područja statike u progmaskom jeziku Visual Basic

Fumić, Robert

Undergraduate thesis / Završni rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, Faculty of Humanities and Social Sciences / Sveučilište u Rijeci, Filozofski fakultet u Rijeci**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:186:558821>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-02-26**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Humanities and Social Sciences - FHSSRI Repository](#)



SVEUČILIŠTE U RIJECI
FILOZOFSKI FAKULTET U RIJECI
ODSJEK ZA POLITEHNIKU

**IZRADA PROGRAMA ZA PROBLEM GREDNI NOSAČ IZ
PODRUČJA STATIKE U PROGRAMSKOM JEZIKU VISUAL
BASIC**

ZAVRŠNI RAD

Robert Fumić

Rijeka, 2017.

SVEUČILIŠTE U RIJECI
FILOZOFSKI FAKULTET U RIJECI
ODSJEK ZA POLITEHNIKU

**IZRADA PROGRAMA ZA PROBLEM GREDNI NOSAČ IZ
PODRUČJA STATIKE U PROGRAMSKOM JEZIKU VISUAL
BASIC**

ZAVRŠNI RAD

Kolegij: Programski jezici
Mentor: Doc. dr. sc. Marko Maliković
Student: Robert Fumić
Studijski program: Politehnika
JMBAG: 0009072197

Rijeka, rujan 2017.

FILOZOFSKI FAKULTET U RIJECI

Odsjek za Politehniku

U Rijeci, 21. ožujka 2017. godine

ZADATAK ZA ZAVRŠNI RAD

(na sveučilišnom preddiplomskom studiju politehnike)

Pristupnik: **Robert Fumić**

Naslov završnog rada: **Izrada programa za problem gredni nosač iz područja statike u programskom jeziku Visual Basic**

Kratak opis zadatka: Opišite problem gredni nosač iz područja statike za koje ćete izraditi program. Opišite ulazne i izlazne podatke programa. Opišite osnovna svojstva programskog jezika Visual Basic, a detaljno opišite klase, objekte, svojstva, metode, varijable i druge elemente programskog jezika Visual Basic koje koristite u programu. Opišite rješenja koja ste dobili iz programa. Priložite programski kôd u programskom jeziku Visual Basic.

Zadatak uručen pristupniku: **21. ožujka 2017. godine**

Ovjera prihvaćanja završnog rada od strane mentora: _____

Završni rad predan: _____

Datum obrane završnog rada: _____

Članovi ispitnog povjerenstva:

1. predsjednik -
2. mentor - Doc. dr. sc. Marko Maliković
3. član -

Konačna ocjena: _____

Mentor

Doc. dr. sc. Marko Maliković

Student: Robert Fumić

Studijski program: Politehnika

JMBAG: 0009072197

IZJAVA

Kojom izjavljujem da sam završni rad s naslovom IZRADA PROGRAMA ZA PROBLEM GREDNI NOSAČ IZ PODRUČJA STATIKE U PROGRAMSKOM JEZIKU VISUAL BASIC izradio samostalno pod mentorstvom Doc. dr. sc. Marko Maliković

U radu sam primijenio/la metodologiju znanstvenoistraživačkog rada i koristio/la literaturu koja je navedena na kraju završnog rada. Tuđe spoznaje, stavove, zaključke, teorije i zakonitosti koje sam izravno ili parafrazirajući naveo u završnom radu na uobičajen, standardan način citirao sam i povezao s fusnotama i korištenim bibliografskim jedinicama. Rad je pisan u duhu hrvatskoga jezika.

Suglasan sam s objavom završnog rada na službenim stranicama Fakulteta.

Student

.....

Robert Fumić

Sažetak

U ovom radu prikazano je kreiranje programa za problem gredni nosač iz područja statike. Rješenje problema je napisano u programskom jeziku Visual Basic, prvenstveno zbog jednostavnosti sintakse programskog jezika Visual Basic kao i zbog mogućnosti crtanja potrebnih grafova za vizualizaciju dobivenih rezultata.

Testiranje izvođenja programa je obavljeno u razvojnoj okolini Microsoft Visual Studio 2015 unutar koje je program i kreiran. Rezultati koje sam dobio rješavanjem zadataka ručno jednaki su rješenjima programa.

Ključne riječi:

- Program
- Gredni nosač
- Statika
- Visual Basic

SADRŽAJ

| | |
|---|----|
| 1. UVOD | 1 |
| 2. PROBLEM GREDNI NOSAČ | 2 |
| 3. PROGRAMSKI JEZIK VISUAL BASIC | 4 |
| 4. ULAZNI I IZLAZNI PODACI | 6 |
| 4.1. ULAZNI PODACI | 7 |
| 4.2. IZLAZNI PODACI | 8 |
| 5. KONTROLE, SVOJSTVA, METODE I VARIJABLE | 10 |
| 5.1. KONTROLE I NJIHOVA SVOJSTVA | 10 |
| 5.2. METODE | 13 |
| 6. OPIS RJEŠENJA PROBLEMA | 21 |
| 6.1 RJEŠENJE PROBLEMA | 21 |
| 6.2. EKSPERIMENTALNI REZULTATI | 22 |
| 7. PROGRAMSKI KÔD | 26 |
| ZAKLJUČAK | 57 |
| Literatura | 58 |

1. UVOD

Koristeći besplatne on-line alate poput SkyCiv Cloud Engineering Softwera, koji u svojoj besplatnoj verziji služe kao kalkulatori za jednostavne gredne nosače, naišao sam na problem nemogućnosti unosa opterećenja pod kutom različitim od 90° , kao i nemogućnost unosa više od dvije koncentrirane sile. Stoga sam motiviran problemom koji studenti imaju sa statikom i rješavanjem zadataka koji su temelj tehničke struke odlučio izraditi program koji će za uneseni statički određen nosač vratiti numerička rješenja i grafički prikazati opterećenja nosača.

Program je kreiran u programskom jeziku Visual Basic koristeći razvojnu okolinu Microsoft Visual Studio 2015 u kojemu je programiranje pogonjeno događajima. Visual Basic nudi mogućnost izračuna potrebnih jednadžbi uvjeta ravnoteže za ostvarenje zadatka čime dolazimo do iznosa reakcijskih sila za odabrane oslonce i iznose momenta fleksije za svaku točku u kojoj se nalazi opterećenje.

Također ovaj programski jezik omogućava programiranje grafičkog prikaza. Koristeći tu mogućnost ostvario sam prikaz grafikona dobivenih iz jednadžba ravnoteže za opterećenja koja imaju komponente na X osi nosača i za opterećenja koja imaju komponente na Y osi nosača. A koristeći iznose momenata fleksije program kreira i grafikon momenata fleksije.

2. PROBLEM GREDNI NOSAČ

Gredni nosači pripadaju statici iz područja fizike, sastavni su dio mehaničkih konstrukcija. Gredni nosači su sklopovi koji kada ih opteretimo prenose ta opterećenja na oslonce. Oslonci mogu biti:

- Pomični
- Nepomični
- Uklještenje

Ovisno o tipu oslonca, isti poprimaju vrijednosti pod opterećenjem. U slučaju pomičnog oslonca kojeg možemo zamisliti kao valjak na koji oslonimo nosač, javlja se reakcijska sila koja ima komponentu na ordinati, odnosno Y osi. Kada bi pomični oslonac opteretili silom koja ima komponentu na apscisi, odnosno X osi, naš nosač bi samo skliznuo s oslonca što je nedopustivo za područje kojim se bavi ovaj rad. Kod nepomičnog oslonca javljaju se reakcijske sile na apscisi i na ordinati. Prilikom uklještenja oslonac poprima reakcije na apscisi, ordinati i još se javlja reakcijski moment koji prikazuje djelovanje sile na kraku. Nosače možemo opteretiti:

- Silama
- Kontinuiranim opterećenjima
- Momentima

Pozicija sila na nosaču je određena jednom točkom na udaljenosti od lijevog kraja nosača, obično izražena u metrima. Može imati jednu komponentu samo po Y osi ako je kut između sile i nosača 90° ili samo po X osi ako je opterećenje paralelno s nosačem odnosno za kut od 0° ili 180° . Ako je kut iz intervala $>0^\circ$ i $<90^\circ$ ili pak $>90^\circ$ i $<180^\circ$ tada se javljaju komponente na apscisi kao i na ordinati. Obično ih izražavamo u Newtonima [N] ili kiloNewton[kN]. Kontinuirana opterećenja karakteriziraju dvije različite točke, na udaljenosti od lijevog kraja nosača, također izraženim u metrima. Između tih točaka, nalazi se kontinuirano opterećenje koje ima jednu komponentu po Y osi izraženu u Newtonima po metru [N/m] ili kiloNewtonima po metru [kN/m]. Momenti su također karakterizirani udaljenošću od lijevog kraja nosača, oni nastoje zarotirati nosač, a intenzitet im je izražen u Newtonmetrima [Nm] ili kiloNewtonmetrima [kNm]. Prilikom rješavanja ovakvih statičkih problema treći Newtonov

zakon nalaže da je suma svih sila na apscisi, kao i suma svih sila na ordinati te suma momenata jednaka nuli.

3. PROGRAMSKI JEZIK VISUAL BASIC

Programski jezik Visual Basic razvila je tvrtka Microsoft, te zbog toga velik broj korisnika platforme Windows koristi Visual Basic za razvoj raznih programa i aplikacija. Visual Basic nastao je na principu programskog jezika Basic (Akronim za **B**eginner's **A**ll-purpose **S**ymbolic **I**nstruction **C**ode) [1.p.53]. Visual Basic pripada pod RAD (Rapid Application Development) što znači da je zamišljen kao alat za brzu izradu programa. Ta brzina ostvarena je dodavanjem gotovih elemenata koji su ugrađeni u sustav Windows. To su GUI (Graphical user interface) koji korisniku omogućuju interakciju s računalom na način da za željenu akciju ostvaruju klik na gumb, ikonu ili slično umjesto čitanja samih linija teksta kako je to ostvareno u TUI (Text-based user interface) programima.

Kod Visual Basica proces programiranja svodi se prvenstveno na pisanje kôda, svaki objekt kojeg vidimo na ekranu potrebno je posebno programirati u kôdu [1.p.53].

Visual Basic je objektno orijentiran, a programiranje je pogonjeno događajima, što znači ako imamo objekt Gumb koji je instanca klase Button koji sadrži metodu za ispis "Pozdrav", prilikom događaja klika na taj gumb na ekran se ispisuje Pozdrav. Metoda koja ispisuje "Pozdrav" bit će pozvana ako i samo ako korisnik obavi događaj klika. Svaki objekt koji možemo koristiti u ovom programskom jeziku ima svoj skup osobina koje možemo mijenjati po želji na način da odaberemo željeni objekt i u prozoru svojstva (eng. Properties) izmijenimo željene parametre. Prilikom izrade programa odaberemo Gumb, koji po principu dovući i ostavi (engl. Drag'n'Drop) dovučemo i ostavimo na našoj radnoj površini (engl. Form) i prilikom dvostukog klika otvara se prozor za pisanje kôda u koji upisujemo metode koje će se izvršiti kada se ostvari klik na gumb.

- **Programiranje pogonjeno događajima** – ideja: da se metoda izvrši ako korisnik želi da metoda bude izvršena, odnosno ako korisnik izvrši operaciju koja potiče aktivaciju metoda nekog objekta (npr. klik).
- **Objektno orijentiran pristup** – ideja: modelirati probleme i situacije iz stvarnog života pomoću objekata. Pri tome je važno što objekt radi, a ne kako on to radi. To omogućava da se pojedini objekt može po potrebi izbaciti i zamijeniti drugim koji će jednu te istu zadaću obaviti na uspješniji način.

Prednosti jezika Visual Basic su jednostavnost korištenja, jednostavno usvajanje kroz rad u razvojnoj okolini Microsoft Visual Studio, velik broj gotovih objekata, sve što vidimo u windows okruženju moguće je programirati u Visual Basic-u. Nedostatak je nemogućnost korištenja pokazivača što uzrokuje lošije upravljanje memorijom [1.p.54].

4. ULAZNI I IZLAZNI PODACI

Ulazno izlazni podaci programa ostvaruju komunikaciju između čovjeka i računala. Da bi program bio svrsishodan mora ostvariti komunikaciju putem korisnikova unosa i programskog ispisa [9]. Kako bi to postigli potrebno je u programskom jeziku Visual Basic odgovarajućom sintaksom deklarirati varijable odnosno rezervirati prostor u memoriji za podatke koje korisnik unosi te rezervirati prostor za rezultate koji se nakon izvršavanja ključnih algoritama pohranjuju i ispisuju. U ovom programu neće biti sadržani samo numerički rezultati programa već će program koristeći algoritam za crtanje kreirati i vizualne izlazne podatke.

Sve naredbe u računalu se izvode na strojnom jeziku. Strojni jezik upravlja sa podacima prikazanim u binarnom obliku, što je za čovjeka teško razumljivo i apstraktno. Stoga su nastali viši programski jezici gdje se naredbe za komunikaciju s računalom pišu preko slova i brojeva [1.p.75]. Varijable možemo shvatiti kao kontejnere za numeričke ili abecedne podatke. U kreiranju programa za naš problem koristio sam iduće varijable (tablica 4.1.).

Tablica4.1. Prikaz korištenih varijabla

| Naziv | Vrsta podatka | Vrijednost |
|---------|--------------------|--|
| Boolean | Logička vrijednost | True/False [1bit] |
| Single | Realni brojevi | $\pm 3.4028235E\pm 38$ [32bita] |
| Double | Realni brojevi | $\pm 1.79769313486231570E+308$ [64bita] |
| String | Znakovi | Od 1 do 65400 znakova |

4.1. ULAZNI PODACI

Ulazni podaci su jedan od ključnih djelova ovog programa. Pomoću njih korisnik definira veličinu nosača, vrste oslonaca, vrste opterećenja i njihove pozicije. Na prvoj kartici (eng. Tab) ovog programa unosimo veličinu nosača i vrste oslonaca i njihove pozicije. Tipovi podataka koje sam koristio su double koji je u mogućnosti poprimiti vrijednost brojeva s decimalnom točkom kao i cijele brojeve, te boolean koji može poprimiti vrijednost 1 i 0 odnosno istina ili laž. Prikazani u tablici (Tablica 4.1.1)

Tablica 4.1.1. Ulazni podaci

| Naziv varijable | Tip podatka |
|--------------------------|-------------|
| duljina_nosaca | double |
| duljina_prepusta | double |
| pomicni_oslonac_lijevo | boolean |
| pomicni_oslonac_desno | boolean |
| nepomicni_oslonac_lijevo | boolean |
| nepomicni_oslonac_desno | boolean |
| ukljestenje | boolean |
| prepust_lijevo | boolean |
| prepust_desno | boolean |
| prepust_obje_strane | boolean |

Na drugoj kartici (eng. Tab) vršimo unos opterećenja. Kako bi to učinkovito obavili, pojavila se potrebnja za složenijim tipom podatka, odnosno strukturama. Strukture su složeni tipovi podataka koji mogu sadržavati više različitih tipova podataka, a tvori ih programer. U ovom programu postoje tri strukture, a to su sila koja se sastoji od intenziteta, pozicije i kuta. Kontinuirano koja se sastoji od intenziteta, pozicije početka i svršetka opterećenja. Moment koja se sastoji od intenziteta, pozicije i smjera. Prikazano u tablici (Tablica 4.1.2).

Tablica 4.1.2. Strukture

| Naziv strukture | Naziv člana strukture | Tip podatka |
|-----------------|------------------------|-------------|
| sila | intenzitet | double |
| | pozicija | double |
| | kut | double |
| kontinuirano | intenzitet | double |
| | pozicija_od | double |
| | pozicija_do | double |
| moment | intenzitet | double |
| | pozicija | double |
| | smjer_kazaljke_na_satu | boolean |

4.2. IZLAZNI PODACI

Izlazni podaci su ono što čini program korisnim. U izlaznim podacima su sadržana rješenja problema koje nam daje naš program pomoću ispravno unesenih ulaznih podataka i odgovarajućih algoritama. U ovom programu izlazni podaci nalaze se na trećoj kartici (eng. Tab) i prikazani su na dva načina, numerički i grafički. Numerički rezultati reakcijskih sila pohranjeni su u varijable tipa double (Tablica 4.2.1) i pomoću objekta oznaka (eng. Label) ispisani na radnu površinu (engl. Form) programa. Za momente fleksije korištena je struktura momenti koja ima mogućnost pohranjivanja intenziteta momenta fleksije te njegove udaljenosti od lijevog kraja nosača. (Tablica 4.2.2)

Tablica 4.2.1. Varijable za pohranu izlaznih podataka reakcije oslonaca

| Naziv varijable | Tip podatka |
|-----------------|-------------|
| reakcijaAx | double |
| reakcijaBx | double |
| reakcijaAy | double |
| reakcijaBy | double |
| momentA | double |
| pozicijaA | double |
| pozicijaB | double |

Tablica 4.2.2. Struktura za pohranu momenata fleksije

| Ime strukture | Ime člana strukture | Tip podatka |
|---------------|---------------------|-------------|
| momenti | intenzitet | double |
| | pozicija | double |

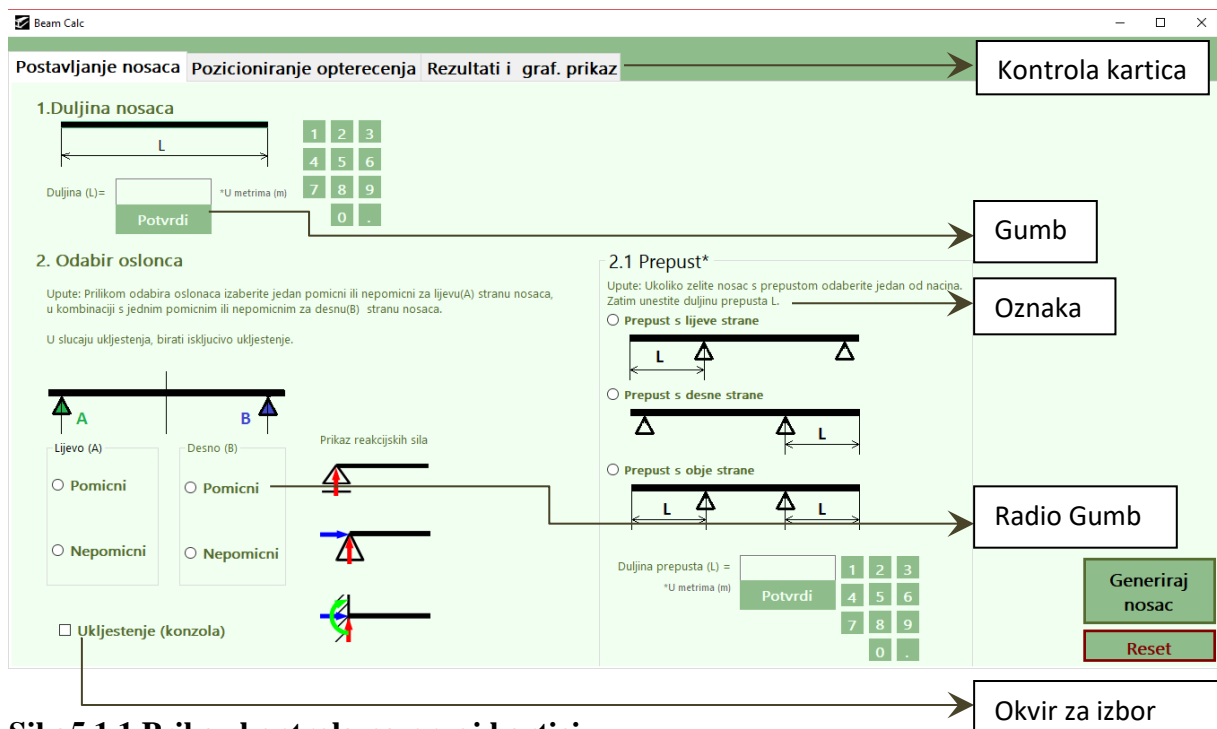
Grafički rezultati prikazani su pomoću algoritma za crtanje na objektu PictureBox naziva „GrafRezultati“ koji se također nalazi na trećoj kartici forme, pozicioniran lijevo od numeričkih rješenja. Kako bi postigli crtanje po objektu „GrafRezultati“ bilo je potrebno deklarirati objekt „myGraphics“ koji je instanca klase Graphics. Za sam postupak crtanja potrebno je još deklarirati objekte „PenGreda“ i „PenGraf“ koji su instance klase Pen i myFont kao instancu klase Font kojima crtamo redom nosač, oslonce, opterećenja, grafikone i obogaćujemo ih ispisom unesenih ulaznih podataka te nekim međurezultatima. Prilikom pisanja algoritama za vizualni prikaz rezultata, naišao sam na problem. Kada sam u For petljama iterirao kroz tip podatka double između 20-30% obrađenih podataka, rezultati su počeli biti nepredvidivi i netočni. Na službenim stranicama kompanije Microsoft, kod pojašnjenja ovog tipa podataka stoji naznačeno da prilikom rada s istim može doći do neočekivanih rezultata iz razloga što nema precizan prikaz u memoriji, stoga sam bio prisiljen sve varijable koje imaju komponentu na apscisi, odnosno x osi, za potrebe crtanja pretvoriti iz tipa double u tip single koji za razliku od 64bita u memoriji zauzima 32bita, a poput double-a služi za pohranu realnih brojeva.

5. KONTROLE, SVOJSTVA, METODE I VARIJABLE

U ovom poglavlju opisat ću detaljno sve kontrole, svojstva, metode i tipove podatka, kao i ostale elemente korištene u pisanju ovog programa.

5.1. KONTROLE I NJIHOVA SVOJSTVA

Ako promotrimo sliku (slika5.1.1) , odnosno prvu karticu programa možemo uočiti kontrole: Kontrola kartica (engl. TabControl), Gumb (engl. Button), Radio gumb (engl. RadioButton), Okvir za izbor (engl. CheckBox) i Oznaku (engl. Label).



Sika5.1.1 Prikaz kontrola na prvoj kartici

Kontrola kartica (engl. TabControl) je kontrola kojom se omogućuje korisniku učinkovito i brzo kretanje između više kartica. U ovom programu imamo 3 kartice. U prvoj kartici korisnik unosi veličinu nosača, odabire oslonce i određuje njihovu poziciju. Na drugoj kartici se kreira shematski prikaz nosača kojeg je korisnik kreirao u prethodnoj kartici, te program nudi korisniku da izabere vrstu i iznos opterećenja kojom želi opteretiti kreiran nosač. Kada je to uspješno obavljeno korisnik pristupa trećoj kartici na kojoj se nalaze numerička i grafička rješenja.

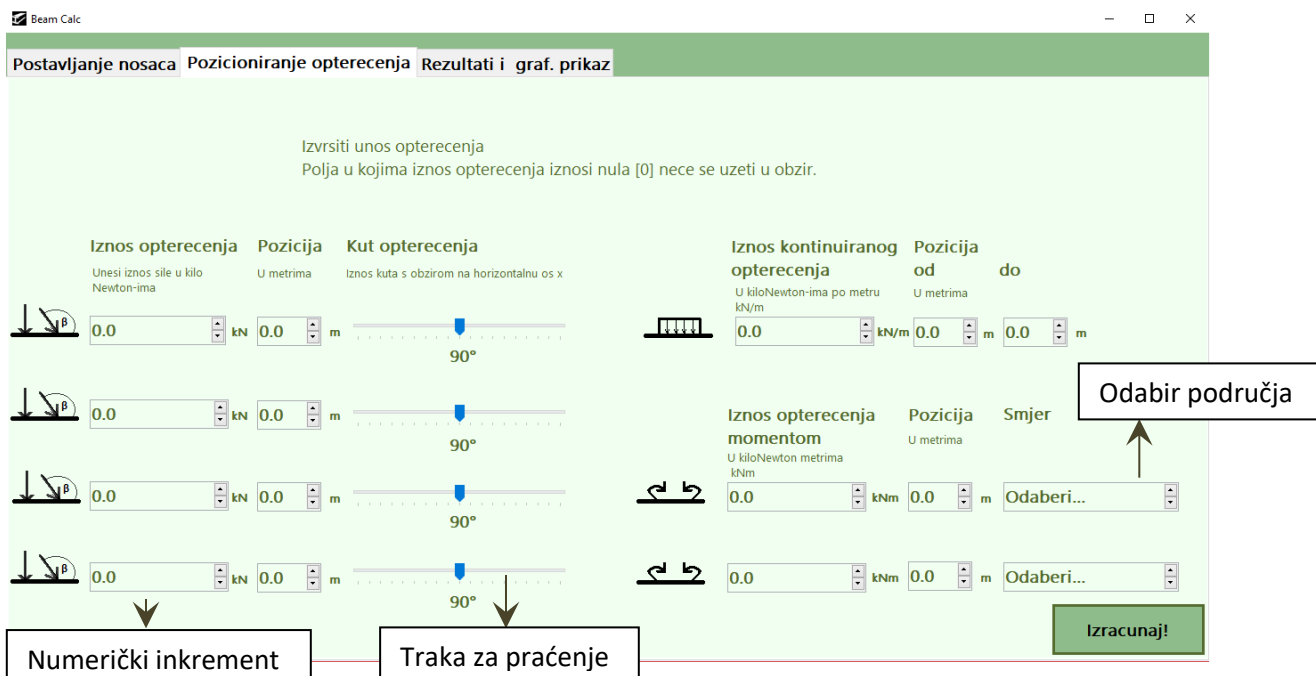
Gumbi (engl. Button) su kontrole koje omogućuju da se započne, prijevremeno prekine ili završi neka radnja [5]. Kada izvršimo klik na gumb pokreće se metoda koja je napisana za tu kontrolu. Svojstva nekog gumba možemo prilagoditi pomoću prozora svojstva (engl. Properties). Unutar prozora sa svojstvima možemo urediti izgled gumba, širinu i visinu, tekst, njegovo ponašanje, dostupnost, možemo mu čak dodjeliti i informaciju o alatu (engl. ToolTip) ne bi li korisniku bilo što jasnije što će izazvati klikom na isti. U ovom programu gumbе koristimo za unos podataka, pokretanje algoritama za kreiranje nosača i izračun reakcijskih sila kao i momenata fleksije, pokretanje algoritma za crtanje grafičkih rezultata i na koncu resetiranje unosa.

Radio gumb (engl. RadioButton) je kontrola koja omogućuje korisniku da odabere jednu opciju od njih nekoliko [5]. Kako bi radio gumbi služili svrsi potrebno je više njih staviti u jednu Grupu (engl. GroupBox). U slučaju kada smo već označili neki radio gumb, ali smo u međuvremenu promijenili mišljenje o odabiru, kada označimo drugi radio gumb onaj prethodni će se automatski odznačiti. Svojstva radio gumba možemo kao i kod običnih gumba mijenjati u svojstvima (engl. Properties). U ovom programu radio gumbе koristimo za odabir oslonaca.

Okvir za izbor (engl. CheckBox) je kontrola koja omogućuje da korisnik odabere neku od opcija ili više njih [5]. Kada korisnik izvrši označavanje (engl. Checked) na okvir za izbor obično se dodjeli neka logička vrijednost nekoj od varijabli. Svojstva okvira za izbor kao i kod prethodnih kontrola možemo izmjenjivati u svojstvima (engl. Properties). U ovom programu okvir za izbor koristimo za odabir uklještenja.

Oznaka (engl. Label) predstavlja standardnu Windows oznaku i obično se koristi za ispis nekog informativnog teksta ili rezultata [5]. I u ovom programu oznake koristimo za informativni tekst i ispis rezultata.

Pogledamo li na sliku (Slika 5.1.2) možemo vidjeti drugu karticu programa na kojoj se nalaze kontrole Numerički inkrement (engl. NumericUpDown), Odabir područja (engl. DomainUpDown), Traka za praćenje (engl. TrackBar)



Slika5.1.2 Prikaz druge kartice programa

Numerički inkrement (engl. NumericUpDown) je kontrola koja omogućuje korisniku unos neke brojčane vrijednosti pomoću dva gumba, od kojih jedan služi za povećanje vrijednosti koja se nalazi unutar Numeričkog inkrementa dok je druga za smanjivanje vrijednosti. Svojstva ove kontrole možemo izmjenjivati na način da mijenjamo njen izgled ili ponašanje ili pak vrijednost i veličinu inkrementa.

Odabir područja (engl. DomainUpDown) je kontrola koja omogućuje korisniku odabir neke od ponuđenih opcija, pomoću dva gumba kreće se kroz izbornik ponuđenih opcija. Svojstva ove kontrole možemo izmjenjivati na način da mijenjamo njen izgled ili ponašanje ili pak količinu i vrstu područja između kojih korisnik bira. U ovom programu odabir područja koristimo prilikom odabira smjera momenta, bilo to u smjeru kazaljke na satu ili obrnuto od smjera kazaljke na satu.

Traka za praćenje (engl. TrackBar) je kontrola koja omogućuje korisniku da podesi neku od numeričkih vrijednosti između minimalne i maksimalne prethodno definirane od strane programera. Svojstva ove kontrole se mogu mijenjati na način da mijenjamo njen izgled ponašanje, minimalne i maksimalne vrijednosti. U ovom programu traku za praćenje

koristimo za određivanje kuta sile, zadana početna vrijednost je 90° a korsnik je može promijeniti u intervalu od 0° do 180° .

Na slici (slika 5.1.3) možemo vidjeti treću karticu programa koja služi za prikaz rezultata, na njoj se nalazi kontrola Okvir za sliku (engl. PictureBox) koja se proteže preko 80% treće kartice, bijele je boje i služi kao podloga za crtanje grafova.



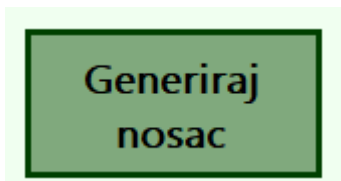
Slika5.1.3 Prikaz treće kartice programa

Okvir za sliku je kontrola koja omogućuje programeru da pomoću slike dočara korisniku potrebno, tako da korisnik nema problema s razumijevanjem koja se informacija od njega iduća traži. Kutiji za sliku možemo dodjeliti bilo koju sliku, a u ovom programu kutiju za sliku koristimo za prikaz veličina koje korisnik unosi i za prikaz grafičkih rezultata. Pri odabiru oslonaca ako korisnik prijeđe pokazivačem miša preko kutije za sliku, dobit će povratnu informacije o osloncu (engl. ToolTip).

5.2. METODE

U objektno orijentiranom programiranju glavnu ulogu imaju objekti koji sadrže podatke i metode. Podaci koje objekt ima predstavljaju njegovo stanje dok se ti podaci pomoću metoda mogu mijenjati i komunicirati s drugim objektima [8]. U ovom dijelu rada

objasniti ću četiri ključne metode koje koristimo u našem programu. Prvu od njih pokrećemo klikom na gumb koji se nalazi na prvoj kartici, u desnom donjem kutu, čiji izgled je prikazan na slici (slika 5.2.1). Prilikom klika na taj gumb izvršava s kôd prikazan na slici (slika 5.2.2).



Slika5.2.1. Gumb za generiranje nosača

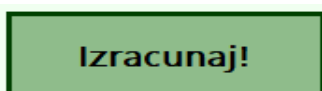
```
230 Private Sub Button23_Click(sender As Object, e As EventArgs) Handles Button23.Click
231     If duljina_nosaca <= 0 Then
232         MsgBox("Niste unjeli duljinu nosaca ili je duljina nosaca negativna, za nastavak potrebno je da duljina bude VECA od 0")
233     ElseIf ukljestenje = False And pomicni_oslonac_lijevo = False And nepomicni_oslonac_lijevo = False Or ukljestenje = False And pomicni_oslonac_desno = False And nepomicni_oslonac_desno = False Or ukljestenje = False And pomicni_oslonac_strane = False And nepomicni_oslonac_strane = False Then
234         MsgBox("Niste odabrali oslonce, za nastavak odaberite 2 oslonca ili ukljestenje")
235     ElseIf duljina_prepusta > duljina_nosaca Then
236         MsgBox("Morate izabrati duljinu prepusta manju od duljine ukupnog nosača")
237     ElseIf PrepustlijevoDesno.Checked And duljina_prepusta * 2 > duljina_nosaca Then
238         MsgBox("Morate izabrati barem 2 puta manju duljinu prepusta od duljine nosača")
239     ElseIf ((prepust_lijevo = True Or prepust_desno = True Or prepust_obje_strane = True) And duljina_prepusta = 0) Then
240         MsgBox("Niste unjeli duljinu prepusta")
241     ElseIf nepomicni_oslonac_lijevo And nepomicni_oslonac_desno Then
242         MsgBox("Aplikacija ne podržava statički neodređene slučajeve, izaberite drugačije oslonce")
243     Else
244         TabControl1.SelectedIndex = TabControl1.SelectedIndex + 1 'klikom na gumb automatski prebacije na iduci tab (Pozicioniranje Sila)
245         If pomicni_oslonac_lijevo = True And pomicni_oslonac_desno = True And prepust_lijevo = False And prepust_desno = False And prepust_obje_strane = False Then
246             PictureBox9.Image = ImageList1.Images(0)
247             VelGrede.Text = duljina_nosaca & "m"
248             TrackBar1.Enabled = False : sila1.kut = 90
249             TrackBar2.Enabled = False : sila2.kut = 90
250             TrackBar3.Enabled = False : sila3.kut = 90
251             TrackBar4.Enabled = False : sila4.kut = 90
252         End If
253         If ukljestenje = True Then
254             PictureBox9.Image = ImageList1.Images(2)
255             VelGrede.Text = duljina_nosaca & "m"
256             TrackBar1.Enabled = True
257             TrackBar2.Enabled = True
258             TrackBar3.Enabled = True
259             TrackBar4.Enabled = True
260         End If
261     End If
End Sub
```

Slika 5.2.2. Programski kôd iza gumba za generiranje nosača

Kada se pokrene ova metoda, ona traži greške u unosu pomoću kôdova koji se nalaze između 230. i 242. linije. To su neispravan unos duljine nosača, neispravan unos oslonaca, unesen prepust veći od ukupne duljine nosača ili ako korisnik odabere statički neodređen slučaj koristeći dva nepomična oslonca. U tim slučajevima program javlja korisniku putem okvira za poruku (engl. MsgBox) da nije moguće izvesti željenu radnju odnosno generiranje nosača. U slučaju kada nije došlo do greške u unosu, ova metoda pomoću kôda na liniji 244 automatski prebacuje na iduću karticu(engl. Tab). Na 245. liniji provjerava se postoje li dva pomična oslonca kako bi mogao prikazati shematski prikaz kreiranog nosača. Kako znamo da u slučaju nosača s pomičnim osloncima ne smijemo opteretiti isti s opterećenjima koja imaju komponentu na apscisi, isključujemo mogućnost mjenjanja kuta i postavljamo fiksnu

vrjednost od 90°, odnosno okomito na nosač. Analogno, uz manje korekcije, ubacujemo kôd za trinaest različitih slučajeva koje ovaj program podržava.

Zatim, korisnik vrši unos opterećenja. Nakon što je korisnik obavio unos informacija o opterećenjima koja djeluju na predhodno kreiran nosač, korisnik klikom na gumb koji se nalazi u desnom donjem kutu čiji izgled je prikazan na slici (slika 5.2.3) pokreće metodu čiji dio je prikazan na slici (slika 5.2.4). Kako se ova metoda proširila preko nekoliko stotina linija kôda razlomio sam ju u 3 ključna trenutka kako bi je mogao što jednostavnije objasniti. Smatram da je ova metoda od velike važnosti programu jer se u njoj dolazi do pronalaska reakcijskih sila, kao i momenta fleksije koji su ključni za crtanje dijagrama opterećenja i dijagrama momenata fleksije.



Slika5.2.3. Gumb za izračun numeričkih rješenja.

```
521 Private Sub Button26_Click(sender As Object, e As EventArgs) Handles Button26.Click
522     TabControl1.SelectedIndex = TabControl1.SelectedIndex + 1 'klikom na gumb automatski prebacije na iduci tab (Pozicioniranje Sila)
523     sila1.kut = (sila1.kut / 180) * System.Math.PI
524     sila2.kut = (sila2.kut / 180) * System.Math.PI
525     sila3.kut = (sila3.kut / 180) * System.Math.PI
526     sila4.kut = (sila4.kut / 180) * System.Math.PI
527     If kontinuirano1.intenzitet <> 0 Then
528         vel_kont = kontinuirano1.pozicija_do - kontinuirano1.pozicija_od
529     End If
530     If DomainUpDown1.Text = "Smjer kazaljke na satu" Then
531         moment1.intenzitet = -(moment1.intenzitet)
532     ElseIf DomainUpDown1.Text = "Smjer obrnut od kazaljke na satu" Then
533         moment1.intenzitet = (moment1.intenzitet)
534     End If
535     If DomainUpDown2.Text = "Smjer kazaljke na satu" Then
536         moment2.intenzitet = -(moment2.intenzitet)
537     ElseIf DomainUpDown2.Text = "Smjer obrnut od kazaljke na satu" Then
538         moment2.intenzitet = (moment2.intenzitet)
539     End If
```

Slika5.2.4.Prvi dio kôda

U prvom dijelu metode dolazimo do vrijednosti s kojima algoritam zna raditi i donosi točne rezultate. Za to je potrebno kutove svih sila pretvoriti u radijane pomoću formule:

$$kut[rad] = \left(\frac{kut[^{\circ}]}{180}\right) * 3.14$$

Nakon što smo to obavili , u liniji 527. provjeravamo postoji li kontinuirano opterećenje, i ako da računamo njegovu duljinu. Dalje u linijama od 530 do 539 podešavamo vrijednosti

momenata s obzirom da li je u smjeru kazaljke na satu ili obrnuto te mu ovisno o slučaju dodjeljujemo određeni predznak. Nastavak metode nalazi se na slici (slika 5.2.5).

```

540 'nepomicni_pomicni_alg
541 If nepomicni_oslonac_lijevo = True And pomicni_oslonac_desno = True Then
542   reakcijaAx = -(sil1.intenzitet * Math.Cos(sil1.kut) + sil2.intenzitet * Math.Cos(sil2.kut) + sil3.intenzitet * Math.Cos(sil3.k
543   reakcijaBy = -(sil1.intenzitet * Math.Sin(sil1.kut) * (sil1.pozicija - pozicijaA) + sil2.intenzitet * Math.Sin(sil2.kut) * (si
544   reakcijaAy = -(sil1.intenzitet * Math.Sin(sil1.kut) + sil2.intenzitet * Math.Sin(sil2.kut) + sil3.intenzitet * Math.Sin(sil3.ki
545 End If
546 'pomicni_nepomicni_alg
547 If pomicni_oslonac_lijevo = True And nepomicni_oslonac_desno = True Then
548   reakcijaBx = -(sil1.intenzitet * Math.Cos(sil1.kut) + sil2.intenzitet * Math.Cos(sil2.kut) + sil3.intenzitet * Math.Cos(sil3.ki
549   reakcijaBy = ((sil1.intenzitet * Math.Sin(sil1.kut) * (sil1.pozicija - pozicijaB) + sil2.intenzitet * Math.Sin(sil2.kut) * (sil
550   reakcijaAy = -(sil1.intenzitet * Math.Sin(sil1.kut) - sil2.intenzitet * Math.Sin(sil2.kut) - sil3.intenzitet * Math.Sin(sil3.ki
551 End If
552 'pomicni_pomicni_alg
553 If pomicni_oslonac_lijevo = True And pomicni_oslonac_desno = True Then
554   reakcijaAx = ((sil1.intenzitet * (sil1.pozicija - pozicijaB) + sil2.intenzitet * (sil2.pozicija - pozicijaB) + sil3.intenzitet *
555   reakcijaAy = (-sil1.intenzitet - sil2.intenzitet - sil3.intenzitet - sil4.intenzitet - kontinuiran01.intenzitet * vel_kont - real
556 End If
557 'ukljestenje
558 If ukljestenje = True Then
559   reakcijaAx = -(sil1.intenzitet * Math.Cos(sil1.kut) + sil2.intenzitet * Math.Cos(sil2.kut) + sil3.intenzitet * Math.Cos(sil3.ki
560   reakcijaAy = -(sil1.intenzitet * Math.Sin(sil1.kut) + sil2.intenzitet * Math.Sin(sil2.kut) + sil3.intenzitet * Math.Sin(sil3.ki
561   momentA = -(sil1.intenzitet * Math.Sin(sil1.kut) * sil1.pozicija + sil2.intenzitet * Math.Sin(sil2.kut) * sil2.pozicija + sila
562 End If
563
564 If reakcijaAx < 0.01 And reakcijaAx > -0.01 Then
565   Label135.Text = "Nema reakcije"
566 Else Label135.Text = (reakcijaAx.ToString("0.000") & " [kN]")
567 End If
568
569 Label136.Text = (reakcijaAy.ToString("0.00") & " [kN]")

```

Slika5.2.5. Izračun reakcija oslonaca

U drugom dijelu ove metode računamo reakcije oslonaca, redom za slučajeve s lijevim nepomičnim i desnim pomičnim osloncem, lijevim pomičnim i desnim nepomičnim te oba pomična i uklještenje. Algoritam radi na principu da prvo nađe reakciju na apscisi pomoću uvjeta da je zbroj svih horizontalnih sila jednak nuli. Zatim računa reakciju na ordinati za nepomični oslonac pomoću uvjeta da je zbroj svih momenata za neku točku jednak nuli. Iz uvjeta da je zbroj svih vertikalnih sila jednak nuli dobijemo i posljednju reakciju. U slučaju uklještenja se javlja još i reakcijski moment koji algoritam računa tako što zbrojimo momente svih opterećenja za točku u kojoj je uklještenje. Nakon računanja reakcija od 564. linije nadalje vršimo ispis rezultata reakcijskih sila na odgovarajuće oznake (engl.Label) na trećoj kartici ovog programa. Kada smo završili računanje reakcijskih sila na red dolazi računanje momenata fleksije za svaku točku u kojoj se nalazi neko opterećenje. To vršimo pomoću algoritma na slici (Slika5.2.6).

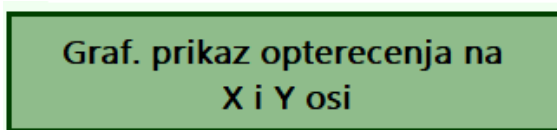
```

926 'racunanje momenata felksije za oslonac A
927 pomocna = 0
928 If ukljestenje = True Then
929     M_fleksijeA = 0 - momentA
930 ElseIf prepust_lijevo = False And prepust_obje_strane = False Then
931     M_fleksijeA = 0
932 ElseIf prepust_lijevo = True Or prepust_obje_strane = True Then
933     If sila1.intenzitet <> 0 And sila1.pozicija <= duljina_prepusta Then
934         pomocna = pomocna + sila1.intenzitet * Math.Sin(sila1.kut) * (duljina_prepusta - sila1.pozicija)
935     End If
936     If sila2.intenzitet <> 0 And sila2.pozicija <= duljina_prepusta Then
937         pomocna = pomocna + sila2.intenzitet * Math.Sin(sila2.kut) * (duljina_prepusta - sila2.pozicija)
938     End If
939     If sila3.intenzitet <> 0 And sila3.pozicija <= duljina_prepusta Then
940         pomocna = pomocna + sila3.intenzitet * Math.Sin(sila3.kut) * (duljina_prepusta - sila3.pozicija)
941     End If
942     If sila4.intenzitet <> 0 And sila4.pozicija <= duljina_prepusta Then
943         pomocna = pomocna + sila4.intenzitet * Math.Sin(sila4.kut) * (duljina_prepusta - sila4.pozicija)
944     End If
945     If kontinuirano1.intenzitet <> 0 And kontinuirano1.pozicija_od < duljina_prepusta Then
946         If kontinuirano1.pozicija_do <= duljina_prepusta Then
947             pomocna = pomocna + kontinuirano1.intenzitet * (kontinuirano1.pozicija_do - kontinuirano1.pozicija_od) * (((kontinuirano
948             ElseIf kontinuirano1.pozicija_do >= duljina_prepusta Then
949                 pomocna = pomocna + kontinuirano1.intenzitet * (duljina_prepusta - kontinuirano1.pozicija_od) * ((duljina_prepusta - kon
950             End If
951         End If
952     If moment1.intenzitet <> 0 And moment1.pozicija <= duljina_prepusta Then
953         pomocna = pomocna - moment1.intenzitet
954     End If
955     If moment2.intenzitet <> 0 And moment2.pozicija <= duljina_prepusta Then
956         pomocna = pomocna - moment2.intenzitet
957     End If
958     M_fleksijeA = pomocna
959 End If

```

Slika 5.2.6 Izračun momenata fleksije za lijevi oslonac

Na slici je prikazan algoritam za računanje momenata fleksije za točku u kojoj se nalazi oslonac A, odnosno lijevi oslonac nosača. Algoritam u 928. liniji provjerava radi li se o uklještenju. Ako je tako, moment fleksije za tu točku poprima vrijednost reakcijskog momenta, zatim u 930. liniji provjeravamo postoji li prepust lijevo od točke za koju računamo moment fleksije, a u slučaju kada ne postoji, moment fleksije automatski poprima vrijednost nula jer znamo da je vrijednost momenta fleksije u krajevima nosača jednak nuli. U suprotnom ispituje se postoji li opterećenje koje je lijevo od našeg oslonca. Ukoliko postoji, računamo moment fleksije za tu komponentu i pribrajamo ga ostatku. Analogno ubacujemo kôd za sva opterećenja koja se mogu naći na nosaču. Nakon uspješno izračunatih reakcijskih sila, momenata i momenata fleksije na red dolazi algoritam za crtanje grafova. Algoritam pokrećemo klikom na objekt koji se nalazi u donjem desnom kutu treće kartice (slika 5.2.7) . Kada izvedemo klik pokreće se algoritam sa slike (slika 5.2.8).



Slika 5.2.7 Gumb za prikaz grafikona opterećenja na x i y osi


```

1221     If reakcijaAx <> 0 Or reakcijaBx <> 0 Then
1222         pomocna = duljina_nosaca * mjeriloX + 20
1223         myGraphics.DrawLine(PenGraf, 20, 220, pomocna, 220) 'crtanje x osi grafa
1224         pomocniString = "[X]"
1225         myGraphics.DrawString(pomocniString, myFont, BrushBrojke, 0, 210)
1226         stopX = 20
1227         stopY = 220
1228         PenGraf = New Pen(Brushes.Blue, 3)
1229     Dim brojac As Single
1230     For brojac = 0 To a Step 1
1231         If brojac = rA And reakcijaAx <> 0 Then
1232             stopX = (brojac) + 20
1233             pomocna = stopY - (reakcijaAx * mjeriloY)
1234             myGraphics.DrawLine(PenGraf, stopX, stopY, stopX, pomocna)
1235             pomocniString = reakcijaAx.ToString("0.000")
1236             myGraphics.DrawString(pomocniString, myFont, BrushBrojke, stopX, pomocna)
1237             stopY = pomocna
1238         End If
1239         If brojac = rB And reakcijaBx <> 0 Then
1240             stopX = (brojac) + 20
1241             pomocna = stopY - (reakcijaBx * mjeriloY)
1242             myGraphics.DrawLine(PenGraf, stopX, stopY, stopX, pomocna)
1243             pomocniString = reakcijaBx.ToString("0.000")
1244             myGraphics.DrawString(pomocniString, myFont, BrushBrojke, stopX, pomocna)
1245             stopY = pomocna
1246         End If
1247         If brojac = s1 And sila1.intenzitet * Math.Cos(sila1.kut) <> 0 Then
1248             stopX = (brojac) + 20
1249             pomocna = stopY - ((sila1.intenzitet * Math.Cos(sila1.kut)) * mjeriloY)
1250             myGraphics.DrawLine(PenGraf, stopX, stopY, stopX, pomocna)
1251             pomocniString = (sila1.intenzitet * Math.Cos(sila1.kut)).ToString("0.000")
1252             myGraphics.DrawString(pomocniString, myFont, BrushBrojke, stopX, pomocna)
1253             stopY = pomocna
1254         End If
1255         If brojac = s2 And sila2.intenzitet * Math.Cos(sila2.kut) <> 0 Then
1256             stopX = (brojac) + 20
1257             pomocna = stopY - ((sila2.intenzitet * Math.Cos(sila2.kut)) * mjeriloY)
1258             myGraphics.DrawLine(PenGraf, stopX, stopY, stopX, pomocna)
1259             pomocniString = (sila2.intenzitet * Math.Cos(sila2.kut)).ToString("0.000")
1260             myGraphics.DrawString(pomocniString, myFont, BrushBrojke, stopX, pomocna)
1261             stopY = pomocna
1262         End If
1263         If brojac = s3 And sila3.intenzitet * Math.Cos(sila3.kut) <> 0 Then
1264             stopX = (brojac) + 20
1265             pomocna = stopY - ((sila3.intenzitet * Math.Cos(sila3.kut)) * mjeriloY)
1266             myGraphics.DrawLine(PenGraf, stopX, stopY, stopX, pomocna)
1267             pomocniString = (sila3.intenzitet * Math.Cos(sila3.kut)).ToString("0.000")
1268             myGraphics.DrawString(pomocniString, myFont, BrushBrojke, stopX, pomocna)
1269             stopY = pomocna
1270         End If
1271         If brojac = s4 And sila4.intenzitet * Math.Cos(sila4.kut) <> 0 Then
1272             stopX = (brojac) + 20
1273             pomocna = stopY - ((sila4.intenzitet * Math.Cos(sila4.kut)) * mjeriloY)
1274             myGraphics.DrawLine(PenGraf, stopX, stopY, stopX, pomocna)
1275             pomocniString = (sila4.intenzitet * Math.Cos(sila4.kut)).ToString("0.000")
1276             myGraphics.DrawString(pomocniString, myFont, BrushBrojke, stopX, pomocna)
1277             stopY = pomocna
1278         End If
1279         pomocna = brojac + 20
1280         myGraphics.DrawLine(PenGraf, stopX, stopY, pomocna, stopY)
1281     Next
1282 End If

```

Slika5.2.8. Prikaz algoritma za crtanje opterećenja x osi

Ovaj algoritam provjerava postoji li horizontalno opterećenje. Ako postoji, pokreće se algoritam za crtanje grafova za horizontalna opterećenja. Kako bi crtanje bilo uspješno potrebno je metodama za crtanje linija proslijediti četiri parametra. Prva dva parametra predstavljaju poziciju na x i y osi od koje započinjemo crtanje, a druga dva predstavljaju poziciju na x i y osi do koje crtamo liniju. Kako računalu u ovom slučaju broj jedan znači jedan piksel, potrebno je uvećati horizontalne veličine za sto, tako da stotinu piksela tvori jedan metar, te bude uočljiviji korisnikovom oku. Crtanje započinjemo s kreiranjem x osi

grafa u kojoj prvo opterećenje započinje i zadnje završava. Zatim ulazimo u for petlju koja prolazi od početka do kraja nosača te svaki puta kada naiđe na opterećenje s komponentom na x osi crta njenu horizontalnu komponentu vertikalno, a u slučaju kada nema opterećenja na x osi crta horizontalnu liniju. Ovaj algoritam je još obogaćen ispisom crtanih vrijednosti pored odgovarajućih linija. Algoritam za crtanje opterećenja na Y osi radi na istom principu. Za crtanje momenata fleksije koristimo idući algoritam. (Slika 5.2.9)

```

1667 For i = 1 To 11 Step 1
1668     If moments(i).pozicija = sila1.pozicija Or moments(i).pozicija = sila2.pozicija Or moments(i).pozicija = sila3.pozicija Or moments(i).pozicija = sila4.pozicija
1669         pomocna = moments(i).pozicija * mjeriloX + 20
1670         pomocna1 = -moments(i).intenzitet * mjeriloY + 330
1671         Dim trenutniMax As Single
1672         Dim pozTrenutnogMax As Single
1673         Dim stvarniMax As Single
1674         Dim pozStvarnogMax As Single
1675         Dim tenzijaKrivulje
1676         stvarniMax = iznosMax
1677         pozStvarnogMax = mjestoMax
1678         trenutniMax = 0
1679         pozTrenutnogMax = 0
1680         Dim max As Single
1681         Dim maxpoz As Single
1682         If kontinuirano1.pozicija_od <= moments(i - 1).pozicija And kontinuirano1.pozicija_do >= moments(i).pozicija And kontinuirano1.intenzitet <> 0 Then
1683             If pozStvarnogMax > (moments(i).pozicija * mjeriloX + 20) Or pozStvarnogMax < (moments(i - 1).pozicija * mjeriloX + 20) Then
1684                 pozTrenutnogMax = ((moments(i).pozicija + moments(i - 1).pozicija) / 2) * mjeriloX + 20
1685                 Dim dio = moments(i).pozicija - moments(i - 1).pozicija
1686                 Dim temp As Double = (pozTrenutnogMax - 20) / mjeriloX
1687                 computeMoment(temp, trenutniMax)
1688                 If moments(i - 1).pozicija = moments(i).pozicija Then
1689                     max = pomocna1
1690                 Else max = 330 - trenutniMax * mjeriloY
1691                 End If
1692                 maxpoz = pozTrenutnogMax
1693             Else
1694                 max = 330 - stvarniMax * mjeriloY
1695                 maxpoz = pozStvarnogMax
1696             End If
1697             If maxpoz >= (((moments(i).pozicija + moments(i - 1).pozicija) * mjeriloX) / 2) Then
1698                 tenzijaKrivulje = ((moments(i).pozicija * mjeriloX + 20) - maxpoz) / ((moments(i).pozicija - moments(i - 1).pozicija) * mjeriloX)
1699             ElseIf maxpoz < (((moments(i).pozicija + moments(i - 1).pozicija) * mjeriloX) / 2) Then
1700                 tenzijaKrivulje = (maxpoz - (moments(i - 1).pozicija * mjeriloX + 20)) / ((moments(i).pozicija - moments(i - 1).pozicija) * mjeriloX)
1701             End If
1702             Dim points() As Point = {New Point(stopX, stopY), New Point(maxpoz, max), New Point(pomocna, pomocna1)}
1703             myGraphics.DrawCurve(PenGraf, points, tenzijaKrivulje)
1704             myGraphics.DrawString(((330 - max) / mjeriloY).ToString("0.00"), myFont, BrushBrojke, maxpoz, max - 10)
1705         Else
1706             myGraphics.DrawLine(PenGraf, stopX, stopY, pomocna, pomocna1)
1707         End If
1708         pomocniString = moments(i).intenzitet.ToString("0.000")
1709         myGraphics.DrawString(pomocniString, myFont, BrushBrojke, pomocna, pomocna1 - 20)
1710         stopX = pomocna
1711         stopY = pomocna1
1712         If moments(i).pozicija = moment1.pozicija Then
1713             If moment1.pozicija = sila1.pozicija Xor moment1.pozicija = sila2.pozicija Xor moment1.pozicija = sila3.pozicija Xor moment1.pozicija = sila4.pozicija X
1714             Else
1715                 If moment1.pozicija = duljina_nosaca Then
1716                     Else
1717                         pomocna1 = stopY + moment1.intenzitet * mjeriloY
1718                         myGraphics.DrawLine(PenGraf, stopX, stopY, stopX, pomocna1)
1719                         stopY = pomocna1
1720                     End If
1721                 End If
1722             End If
1723         If moments(i).pozicija = moment2.pozicija Then
1724             If moment2.pozicija = sila1.pozicija Xor moment2.pozicija = sila2.pozicija Xor moment2.pozicija = sila3.pozicija Xor moment2.pozicija = sila4.pozicija X
1725             Else
1726                 If moment2.pozicija = duljina_nosaca Then
1727                     Else
1728                         pomocna1 = stopY + moment2.intenzitet * mjeriloY
1729                         myGraphics.DrawLine(PenGraf, stopX, stopY, stopX, pomocna1)
1730                         stopY = pomocna1
1731                     End If
1732                 End If
1733             End If
1734         End If
1735     Next

```

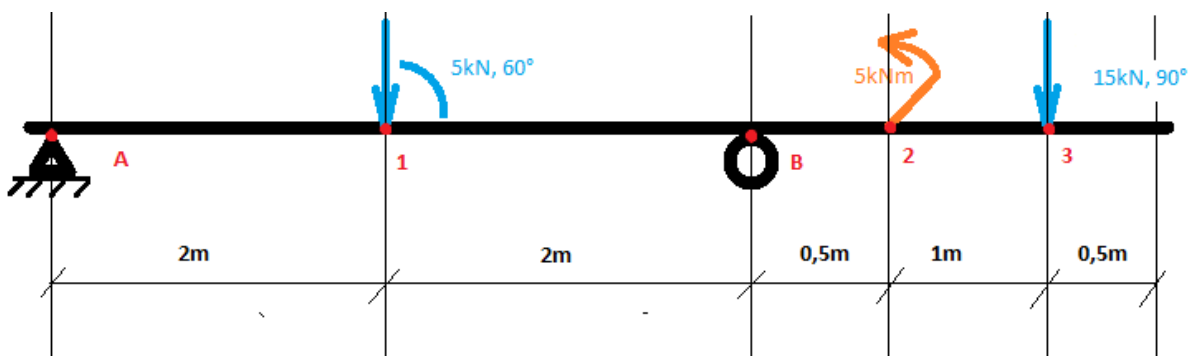
Slika5.2.9. Prikaz algoritma za crtanje momenata fleksije

Kod crtanja momenata fleksije pristupamo problemu na drugačiji način. Prethodno smo, prilikom računanja numeričkih rezultata, izračunali moment fleksije za svaku točku u kojoj

postoji opterećenje. Ti momenti se trenutno nalaze u jednodimenzionalnom polju tipa strukture „momenti“. Ovo jednodimenzionalno polje ima točno dvanaest članova za koje pretpostavljamo da su nasumično poredani jer se od korisnika ne zahtijeva da unos opterećenja vrši s lijeva na desno, već ih može unositi kojim god redoslijedom želi. Nakon tog to polje se sortira algoritmom za sortiranje „Bubble Sort“, koji je po svojoj brzini jedan od sporijih algoritama za sortiranje, ali to ne igra veliku ulogu u brzini izvođenja programa jer sortira samo dvanaest vrijednosti. Prisjetimo se na trenutak da struktura „momenti“ ima članove „pozicija“ i „intenzitet“, koji u biti predstavljaju koordinate grafa momenata fleksije. Član pozicija predstavlja x koordinatu dok intenzitet predstavlja y koordinatu te tih dvanaest vrijednosti tvori poligon točaka grafa. Da bi crtanje bilo uspješno potrebno je samo ravnim linijama spojiti te ključne točke sortiranog niza, odnosno zakrivljenim linijama ako je u pitanju kontinuirano opterećenje.

6. OPIS RJEŠENJA PROBLEMA

U ovom poglavlju prikazano je rješenje jednog jednostavnijeg problema kroz proračun koji ćemo zatim usporediti s programskim rješenjima za isti slučaj (slika 6.1).



Slika6.1 Zadatak

6.1 RJEŠENJE PROBLEMA

U točki A, koja se nalazi na početku nosača, postavljen je nepomični oslonac što znači da se i njemu nalaze reakcije na x i y osi. U točki B postavljen je pomični oslonac, a on sadržava reakciju samo na y osi. Izračunajmo ih koristeći uvijete $\sum F_x=0$, $\sum F_y=0$, $\sum M_A=0$.

$$\sum F_x=0 \quad F_{Ax}=F_1 \cdot \cos(60) \quad F_{Ax}=2.5[\text{kN}]$$

$$\sum M_A=0 \quad F_{By} = \frac{F_1 \cdot \sin(60) \cdot 2m + F_2 \cdot \sin(90) \cdot 5.5m - M_1}{4m} \quad F_{By}=21.54[\text{kN}]$$

$$\sum F_y=0 \quad F_{Ay}=F_1 \cdot \sin(60) + F_2 \cdot \sin(90) - F_{By} \quad F_{Ay} = -2,21[\text{kN}]$$

F_{Ax} = x komponenta reakcijske sile oslonca A

F_{Ay} = y komponenta reakcijske sile oslonca A

F_{By} = y komponenta reakcijske sile oslonca B

F_1 = iznos prve sile

F_2 = iznos druge sile

Prilikom proračuna, za reakciju F_{Ay} dobili smo negativan predznak, što znači da smo krivo pretpostavili smjer. Zahvaljujući tom negativnom predznaku i dalje ćemo dobivati ispravne rezultate. Dalje nastavljamo s proračunom momenata fleksije za svaku točku u kojoj se nalaze opterećenja. Na krajevima nosača moment fleksije je uvijek 0.

$$Mf_A=0$$

$$Mf_1=F_{Ay} * 2m = -4,42[kNm]$$

$$Mf_B=F_{Ay} * 4m - F_1 * \sin(60) * 2m = -17,5[kNm]$$

$$Mf_2 \text{ bez momenta}=F_{Ay} * 4,5m - F_1 * \sin(60) * 2,5m + F_{By} * 0,5m = -10[kNm]$$

$$Mf_2 \text{ s momentom} = Mf_2 + M_1 = -15[kNm]$$

$$Mf_3=F_{Ay} * 5,5m - F_1 * \sin(60) * 3,5m + F_{By} * 1,5m - M_1 = 0[kNm]$$

Mf_A = moment fleksije za oslonac A

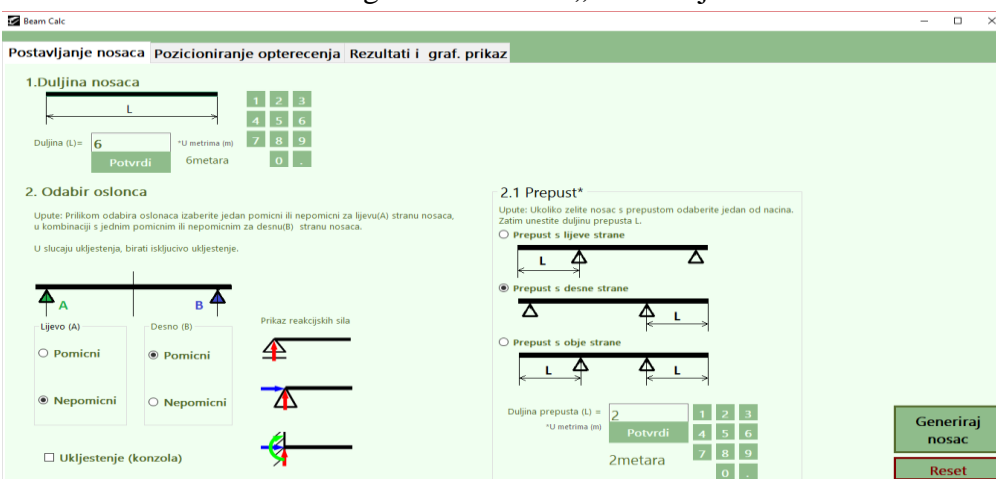
Mf_B = moment fleksije za oslonac B

$Mf_{1,2,3}$ = momenti fleksije za točke 1, 2 i 3

M_1 = Iznos momenta1

6.2. EKSPERIMENTALNI REZULTATI

Kako bi prikazali programska rješenja za ovaj slučaj, trebamo računalu predati ulazne podatke. Prvo odredimo nosač veličine šest metara, pridružimo mu nepomični oslonac u lijevom kraju nosača te pomični u desnom kraju uz prepust od 2 metra (slika6.2.1). Kada smo to obavili izvodimo klik na gumb s tekстом „Generiraj nosac“



Slika6.2.1 Prikaz prve kartice programa

Kada smo kliknuli na gumb s tekстом „Generiraj nosač“, metoda iza tog gumba je prebacila kontrolu na novu karticu programa u kojoj vršimo unos opterećenja, čiji izgled možemo vidjeti na slici (slika 6.2.2).

U polja za prvu silu unosimo redom:

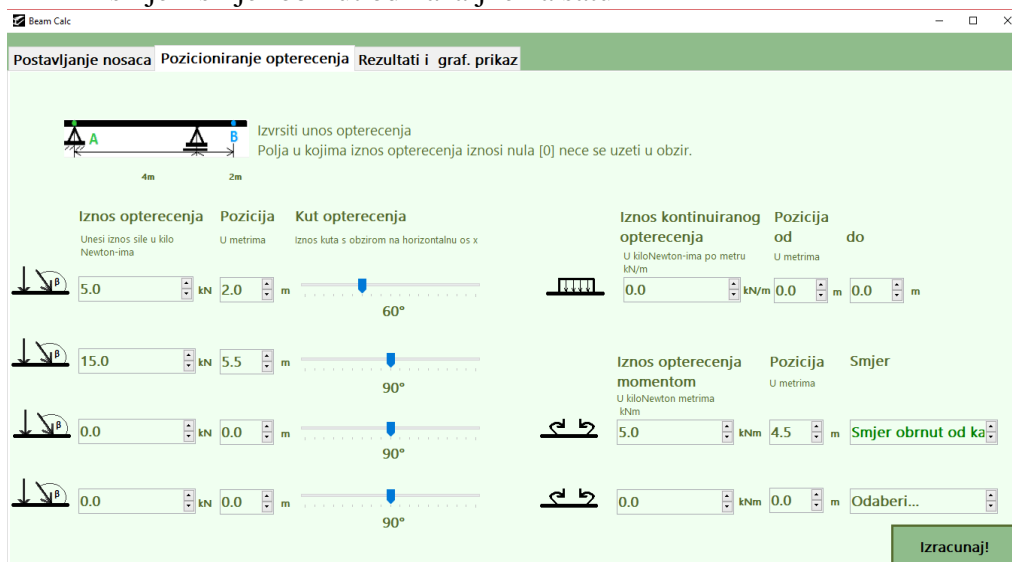
- iznos opterećenja=5[kN]
- pozicija=2[m]
- podesimo kut opterećenja koristeći traku za praćenje (engl.TrackBar) na 60°

U polja za drugu silu unosimo redom:

- iznos opterećenja=15[kN]
- pozicija=5.5[m]
- ostavimo kut opterećenja na pretpostavljenoj vrijednosti od 90°

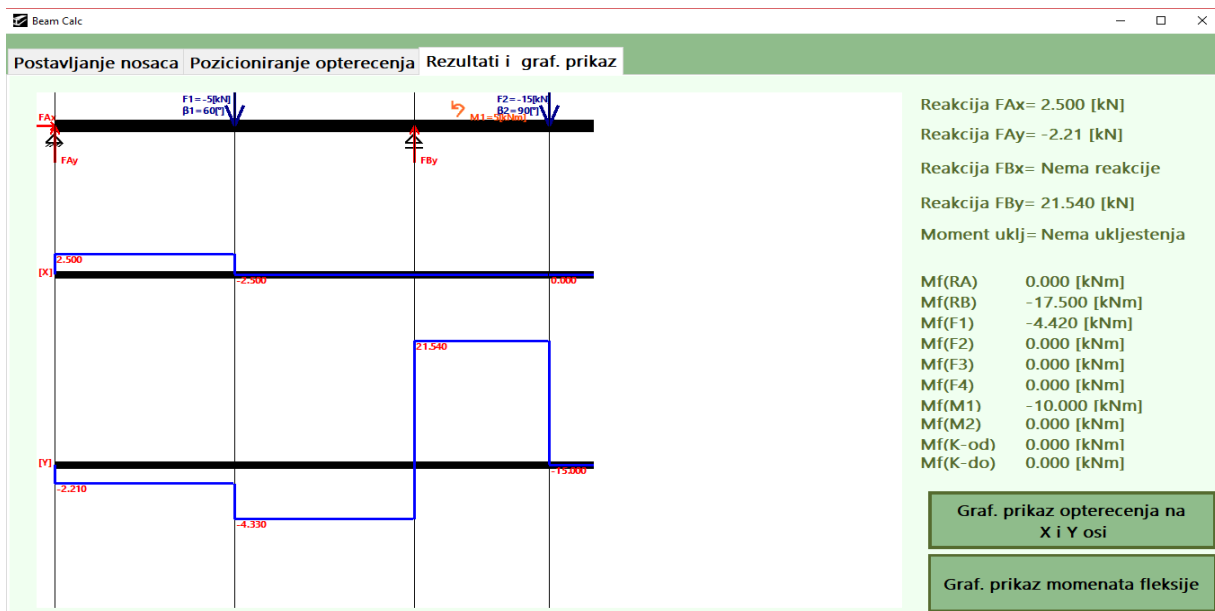
U polja za moment unosimo redom:

- iznos momenta=5[kNm]
- pozicija=4.5[m]
- smjer=smjer obrnut od kazaljke na satu



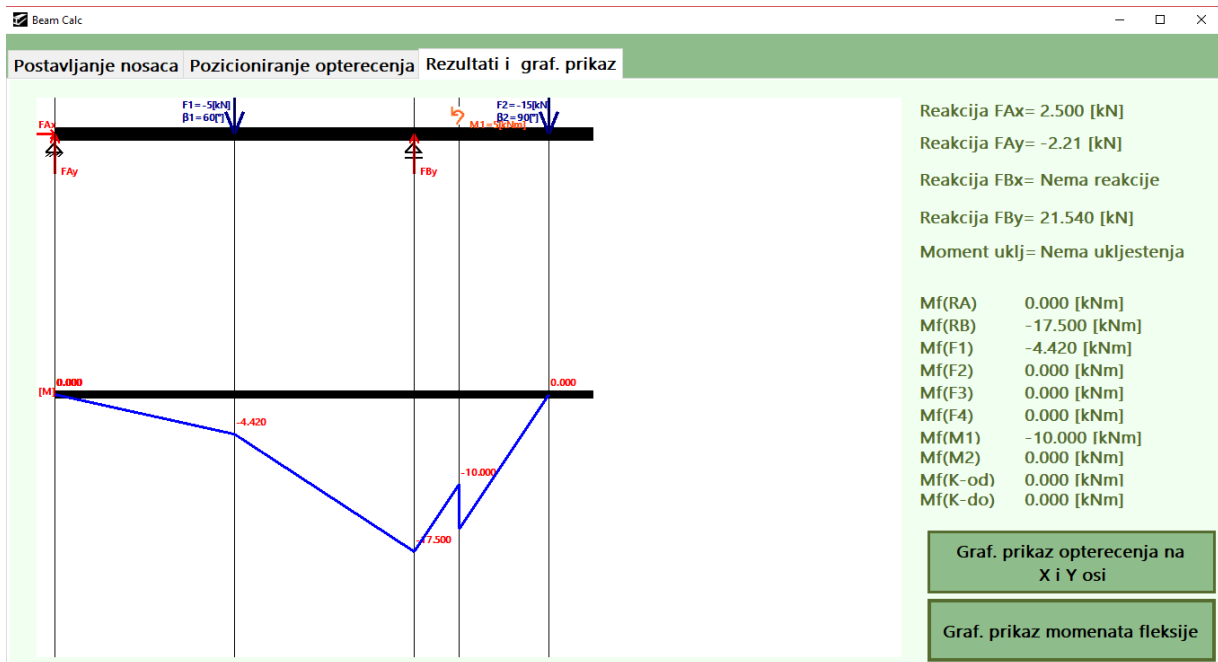
Slika 6.2.2 Prikaz druge kartice programa

Kada smo izvršili unos opterećenja za naš nosač, izvodimo klik na gumb s tekстом „Izračunaj“. Metoda iza tog gumba prebacuje na karticu s rješenjima čiji izgled možemo vidjeti na slici (slika 6.2.3).



Slika6.2.3. Prikaz treće kartice programa s grafikonima za X i Y os

Iz priložene slike možemo vidjeti da nam program daje jednaka rješenja kao i u primjeru koji smo riješili ručno, klikom na gumb s tekстом „Graf. Prikaz opterećenja na X i Y osi“ pokreće se metoda za crtanje grafova koja ostavlja trag opterećenja na x i y. Na gornjem od dva grafa prikazano je opterećenje na x osi nosača, kako samo dvije komponente imaju opterećenje na x osi on je vrlo jednostavan. S lijeva na desno crta iznos reakcije F_{Ax} vertikalno, zatim horizontalnom linijom ide bez promijene sve do opterećenja F_1 gdje crta njenu horizontalnu komponentu vertikalno i dolazi do nulte vrijednosti. Na donjem od dva grafa prikazano je opterećenje y osi nosača te je uočljivo kako se graf mijenja nailaskom na opterećenja. Redom se crta F_{Ay} vertikalno, zatim horizontalnom linijom ide bez promijene sve do opterećenja F_1 gdje ponovo ostavlja trag u obliku vertikalne linije veličine iznosa opterećenja $F_1(\sin 60)$, horizontalnom linijom bez promijene nastavlja sve do F_{By} gdje nanosi njen iznos vertikalno. Na koncu dolazimo horizontalnom linijom u mjesto opterećenja F_2 čiji iznos se nanosi vertikalno te graf dolazi u nultu vrijednost. Na slici (slika6.2.4) pogledajmo graf momenata fleksije.



Slika6.2.4. Prikaz treće kartice programa s grafikonom za momente fleksije

U jednodimenzionalnom polju strukture momenti je pohranjen poligon točaka ovog grafa kojeg jednostavno spojimo odgovarajućim linijama. Moment fleksije u početku nosača iznosi vrijednost 0, što je ujedno i iznos momenta fleksije za oslonac A, u drugoj točki, odnosno, na poziciji sile F_1 moment fleksije iznosi -4.42kNm. Računalo spaja te dvije dočke linijom te nastavlja dalje do oslonca B u čijoj poziciji moment fleksije iznosi -17.5kNm. Nakon tog dolazimo u točku momenta1 u kojoj je iznos momenta fleksije -10kNm, te algoritam za crtanje grafa od te točke nanosi vertikalno iznos momenta u izonsu od 5kNm, čime dolazi do vrijednosti od 15kNm iz ručnog proračuna. Moment fleksije za poziciju sile F_2 iznosi 0kNm, čime graf dolazi u nultu i krajnju vrijednost.

7. PROGRAMSKI KÔD

Slijedi prikaz implementacije programskog kôda za problem gredni nosač.

```
Public Class Form1
    Dim pomocniString As String
    Dim pomocna As Double
    Dim mjestoMax As Single
    Dim iznosMax As Double
    Public duljina_nosaca As Double = 0
    Public duljina_prepusta As Double = 0
    Public pomicni_oslonac_lijevo As Boolean = False
    Public pomicni_oslonac_desno As Boolean = False
    Public nepomicni_oslonac_lijevo As Boolean = False
    Public nepomicni_oslonac_desno As Boolean = False
    Public ukljestenje As Boolean = False
    Public prepust_lijevo As Boolean = False
    Public prepust_desno As Boolean = False
    Public prepust_obje_strane As Boolean = False
    Public razmak_oslonaca As Double
    Public Structure sila
        Public intenzitet As Double
        Public pozicija As Double
        Public kut As Double
    End Structure
    Public sila1 As sila
    Public sila2 As sila
    Public sila3 As sila
    Public sila4 As sila
    Public Structure kontinuirano
        Public intenzitet As Double
        Public pozicija_od As Double
        Public pozicija_do As Double
    End Structure
    Public kontinuirano1 As kontinuirano
    Public vel_kont As Double
    Public Structure moment
        Public intenzitet As Double
        Public pozicija As Double
        Public smjer_kazaljke_na_satu As Boolean
    End Structure
    Public moment1 As moment
    Public moment2 As moment
    Public reakcijaAx As Double
    Public reakcijaBx As Double
    Public reakcijaAy As Double
    Public reakcijaBy As Double
    Public pozicijaA As Double
    Public pozicijaB As Double
    Public momentA As Double
    Public razmakNepom0 As Double
    Public M_fleksijeA As Double
    Public M_fleksijeB As Double
    Public M_fleksijeS1 As Double
    Public M_fleksijeS2 As Double
    Public M_fleksijeS3 As Double
    Public M_fleksijeS4 As Double
    Public M_fleksijeK1 As Double

```

```

Public M_fleksijeK11 As Double
Public M_fleksijeM1 As Double
Public M_fleksijeM2 As Double
Public Structure momenti
    Public intenzitet As Double
    Public pozicija As Double
End Structure
Public moments(12) As momenti
Public max As Double
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    TextBox1.SelectedText = Button1.Text
    TextBox1.Focus()
End Sub
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    TextBox1.SelectedText = Button2.Text
    TextBox1.Focus()
End Sub
Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
    TextBox1.SelectedText = Button3.Text
    TextBox1.Focus()
End Sub
Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
    TextBox1.SelectedText = Button4.Text
    TextBox1.Focus()
End Sub
Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
    TextBox1.SelectedText = Button5.Text
    TextBox1.Focus()
End Sub
Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
    TextBox1.SelectedText = Button6.Text
    TextBox1.Focus()
End Sub
Private Sub Button7_Click(sender As Object, e As EventArgs) Handles Button7.Click
    TextBox1.SelectedText = Button7.Text
    TextBox1.Focus()
End Sub
Private Sub Button8_Click(sender As Object, e As EventArgs) Handles Button8.Click
    TextBox1.SelectedText = Button8.Text
    TextBox1.Focus()
End Sub
Private Sub Button9_Click(sender As Object, e As EventArgs) Handles Button9.Click
    TextBox1.SelectedText = Button9.Text
    TextBox1.Focus()
End Sub
Private Sub Button10_Click(sender As Object, e As EventArgs) Handles Button10.Click
    TextBox1.SelectedText = Button10.Text
    TextBox1.Focus()
End Sub
Private Sub Button11_Click(sender As Object, e As EventArgs) Handles Button11.Click
    TextBox1.SelectedText = Button11.Text
    TextBox1.Focus()
End Sub
Private Sub Button20_Click(sender As Object, e As EventArgs) Handles Button20.Click
    TextBox2.SelectedText = Button20.Text
    TextBox2.Focus()
End Sub
Private Sub Button12_Click(sender As Object, e As EventArgs) Handles Button12.Click
    TextBox2.SelectedText = Button12.Text
    TextBox2.Focus()

```

```

End Sub
Private Sub Button13_Click(sender As Object, e As EventArgs) Handles Button13.Click
    TextBox2.SelectedText = Button13.Text
    TextBox2.Focus()
End Sub
Private Sub Button14_Click(sender As Object, e As EventArgs) Handles Button14.Click
    TextBox2.SelectedText = Button14.Text
    TextBox2.Focus()
End Sub
Private Sub Button15_Click(sender As Object, e As EventArgs) Handles Button15.Click
    TextBox2.SelectedText = Button15.Text
    TextBox2.Focus()
End Sub
Private Sub Button16_Click(sender As Object, e As EventArgs) Handles Button16.Click
    TextBox2.SelectedText = Button16.Text
    TextBox2.Focus()
End Sub
Private Sub Button17_Click(sender As Object, e As EventArgs) Handles Button17.Click
    TextBox2.SelectedText = Button17.Text
    TextBox2.Focus()
End Sub
Private Sub Button18_Click(sender As Object, e As EventArgs) Handles Button18.Click
    TextBox2.SelectedText = Button18.Text
    TextBox2.Focus()
End Sub
Private Sub Button19_Click(sender As Object, e As EventArgs) Handles Button19.Click
    TextBox2.SelectedText = Button19.Text
    TextBox2.Focus()
End Sub
Private Sub Button21_Click(sender As Object, e As EventArgs) Handles Button21.Click
    TextBox2.SelectedText = Button21.Text
    TextBox2.Focus()
End Sub
Private Sub Button22_Click(sender As Object, e As EventArgs) Handles Button22.Click
    TextBox2.SelectedText = Button22.Text
    TextBox2.Focus()
End Sub
Private Sub Button24_Click(sender As Object, e As EventArgs) Handles Button24.Click
    If TextBox1.TextLength = 0 Then 'da li je unesen broj
        MsgBox("Niste unesli broj, pokušajte ponovo!") 'ako nije, korisnik dobije poruku da ponovo unese broj
    Else
        pomocniString = TextBox1.Text 'inače - spremi broj u pomocniString
        pomocniString = pomocniString.Replace(",",".") 'ako se u pomocnomStringu pojavljuje ",", zamjeni ga s "."
        duljina_nosaca = Cdbl(pomocniString) 'funkcijom Cdbl mjenjamo string u double
        Label2.Text = duljina_nosaca & "metara" 'ispis unesene double varijable
    End If
End Sub
Private Sub Button25_Click(sender As Object, e As EventArgs) Handles Button25.Click
    If TextBox2.TextLength = 0 Then
        MsgBox("Niste unesli broj, pokušajte ponovo!")
    ElseIf TextBox2.TextLength > 0 Then
        pomocniString = TextBox2.Text
        pomocniString = pomocniString.Replace(",",".")
        duljina_prepusta = Cdbl(pomocniString)
        If duljina_prepusta >= duljina_nosaca Then
            MsgBox("Prepust ne može biti veći od duljine cijelog grednog nosaca! Pokušajte ponovo.")
            duljina_prepusta = 0
        End If
    End If
    Label19.Text = duljina_prepusta & "metara"

```

```

End Sub
Private Sub RadioButton6_CheckedChanged(sender As Object, e As EventArgs) Handles pomicniA.CheckedChanged
    pomicni_oslonac_lijevo = True
    nepomicni_oslonac_lijevo = False
    konzola.Checked = False
End Sub
Private Sub RadioButton4_CheckedChanged(sender As Object, e As EventArgs) Handles pomicniB.CheckedChanged
    pomicni_oslonac_desno = True
    nepomicni_oslonac_desno = False
    konzola.Checked = False
End Sub
Private Sub RadioButton7_CheckedChanged(sender As Object, e As EventArgs) Handles nepomicniA.CheckedChanged
    nepomicni_oslonac_lijevo = True
    pomicni_oslonac_lijevo = False
    konzola.Checked = False
End Sub
Private Sub RadioButton5_CheckedChanged(sender As Object, e As EventArgs) Handles nepomicniB.CheckedChanged
    nepomicni_oslonac_desno = True
    pomicni_oslonac_desno = False
    konzola.Checked = False
End Sub
Private Sub CheckBox1_CheckedChanged(sender As Object, e As EventArgs) Handles konzola.CheckedChanged
    ukljestenje = True
    pomicniA.Checked = False 'odznaci RadioButton za pomicni A, Analogno ubaceno za sve slucajeve DORADITI OVAJ DIO JOS!!!!!!!
    pomicniB.Checked = False
    nepomicniA.Checked = False
    nepomicniB.Checked = False
    pomicni_oslonac_lijevo = False
    pomicni_oslonac_desno = False
    nepomicni_oslonac_lijevo = False
    nepomicni_oslonac_desno = False
    PrepustLijevo.Checked = False
    PrepustDesno.Checked = False
    PrepustLijevoDesno.Checked = False
    duljina_prepusta = 0
    PrepustBox.Enabled = False
    If konzola.Checked = False Then
        PrepustBox.Enabled = True
    End If
End Sub
Private Sub PrepustLijevo_CheckedChanged(sender As Object, e As EventArgs) Handles PrepustLijevo.CheckedChanged
    prepust_lijevo = True
    prepust_desno = False
    prepust_obje_strane = False
End Sub
Private Sub PrepustDesno_CheckedChanged(sender As Object, e As EventArgs) Handles PrepustDesno.CheckedChanged
    prepust_lijevo = False
    prepust_desno = True
    prepust_obje_strane = False
End Sub
Private Sub PrepustLijevoDesno_CheckedChanged(sender As Object, e As EventArgs) Handles PrepustLijevoDesno.CheckedChanged
    prepust_lijevo = False
    prepust_desno = False
    prepust_obje_strane = True
End Sub
Private Sub Button23_Click(sender As Object, e As EventArgs) Handles Button23.Click
    If duljina_nosaca <= 0 Then
        MsgBox("Niste unjeli duljinu nosaca ili je duljina nosaca negativna, za nastavak potrebno je da duljina bude VECA od 0")
    ElseIf ukljestenje = False And pomicni_oslonac_lijevo = False And nepomicni_oslonac_lijevo = False Or ukljestenje = False And pomicni_oslonac_desno = False And nepomicni_oslonac_desno = False Then

```

```

        MsgBox("Niste odabrali oslonce, za nastavak odaberite 2 oslonca ili uklestenje")
    ElseIf duljina_prepusta > duljina_nosaca Then
        MsgBox("Morate izabrati duljinu prepusta manju od duljine ukupnog nosača")
    ElseIf PrepustLijevoDesno.Checked And duljina_prepusta * 2 > duljina_nosaca Then
        MsgBox("Morate izabrati barem 2 puta manju duljinu prepusta od duljine nosača")
    ElseIf ((prepust_lijevo = True Or prepust_desno = True Or prepust_obje_strane = True) And duljina_prepusta = 0) Then
        MsgBox("Niste unjeli duljinu prepusta")
    ElseIf nepomicni_oslonac_lijevo And nepomicni_oslonac_desno Then
        MsgBox("Aplikacija ne podržava statički neodređene slučajeve, izaberite drugačije oslonce")
    Else
        TabControl1.SelectedIndex = TabControl1.SelectedIndex + 1 'klikom na gumb automatski prebacije na iduci tab (Pozicioniranje
Sila)
        If pomicni_oslonac_lijevo = True And pomicni_oslonac_desno = True And prepust_lijevo = False And prepust_desno = False And
prepust_obje_strane = False Then
            PictureBox9.Image = ImageList1.Images(0)
            VelGrede.Text = duljina_nosaca & "m"
            TrackBar1.Enabled = False : sila1.kut = 90
            TrackBar2.Enabled = False : sila2.kut = 90
            TrackBar3.Enabled = False : sila3.kut = 90
            TrackBar4.Enabled = False : sila4.kut = 90
        End If
        If ukljestenje = True Then
            PictureBox9.Image = ImageList1.Images(2)
            VelGrede.Text = duljina_nosaca & "m"
            TrackBar1.Enabled = True
            TrackBar2.Enabled = True
            TrackBar3.Enabled = True
            TrackBar4.Enabled = True
        End If
        If pomicni_oslonac_lijevo = True And nepomicni_oslonac_desno = True And prepust_lijevo = False And prepust_desno = False And
prepust_obje_strane = False Then
            PictureBox9.Image = ImageList1.Images(3)
            VelGrede.Text = duljina_nosaca & "m"
            TrackBar1.Enabled = True
            TrackBar2.Enabled = True
            TrackBar3.Enabled = True
            TrackBar4.Enabled = True
        End If
        If nepomicni_oslonac_lijevo = True And pomicni_oslonac_desno = True And prepust_lijevo = False And prepust_desno = False And
prepust_obje_strane = False Then
            PictureBox9.Image = ImageList1.Images(4)
            VelGrede.Text = duljina_nosaca & "m"
            TrackBar1.Enabled = True
            TrackBar2.Enabled = True
            TrackBar3.Enabled = True
            TrackBar4.Enabled = True
        End If
        If pomicni_oslonac_lijevo = True And pomicni_oslonac_desno = True And prepust_lijevo = True Then
            PictureBox9.Image = ImageList1.Images(5)
            VelGrede.Text = duljina_nosaca - duljina_prepusta & "m"
            LblPrepustLijevo.Text = duljina_prepusta & "m"
            TrackBar1.Enabled = False
            sila1.kut = 90
            TrackBar2.Enabled = False
            sila2.kut = 90
            TrackBar3.Enabled = False
            sila3.kut = 90
            TrackBar4.Enabled = False
            sila4.kut = 90
        End If
    End If

```

```

If pomicni_oslonac_lijevo = True And pomicni_oslonac_desno = True And prepust_desno = True Then
    PictureBox9.Image = ImagemList1.Images(6)
    VelGrede.Text = duljina_nosaca - duljina_prepusta & "m"
    LblPrepustDesno.Text = duljina_prepusta & "m"
    TrackBar1.Enabled = False
    sila1.kut = 90
    TrackBar2.Enabled = False
    sila2.kut = 90
    TrackBar3.Enabled = False
    sila3.kut = 90
    TrackBar4.Enabled = False
    sila4.kut = 90
End If
If pomicni_oslonac_lijevo = True And pomicni_oslonac_desno = True And prepust_obje_strane = True Then
    PictureBox9.Image = ImagemList1.Images(7)
    VelGrede.Text = duljina_nosaca - duljina_prepusta * 2 & "m"
    LblPrepustLijevo.Text = duljina_prepusta & "m"
    LblPrepustDesno.Text = duljina_prepusta & "m"
    TrackBar1.Enabled = False
    sila1.kut = 90
    TrackBar2.Enabled = False
    sila2.kut = 90
    TrackBar3.Enabled = False
    sila3.kut = 90
    TrackBar4.Enabled = False
    sila4.kut = 90
End If
If pomicni_oslonac_lijevo = True And nepomicni_oslonac_desno = True And prepust_lijevo = True Then
    PictureBox9.Image = ImagemList1.Images(11)
    VelGrede.Text = duljina_nosaca - duljina_prepusta & "m"
    LblPrepustLijevo.Text = duljina_prepusta & "m"
    TrackBar1.Enabled = True
    TrackBar2.Enabled = True
    TrackBar3.Enabled = True
    TrackBar4.Enabled = True
End If
If pomicni_oslonac_lijevo = True And nepomicni_oslonac_desno = True And prepust_desno = True Then
    PictureBox9.Image = ImagemList1.Images(12)
    VelGrede.Text = duljina_nosaca - duljina_prepusta & "m"
    LblPrepustDesno.Text = duljina_prepusta & "m"
    TrackBar1.Enabled = True
    TrackBar2.Enabled = True
    TrackBar3.Enabled = True
    TrackBar4.Enabled = True
End If
If pomicni_oslonac_lijevo = True And nepomicni_oslonac_desno = True And prepust_obje_strane = True Then
    PictureBox9.Image = ImagemList1.Images(13)
    VelGrede.Text = duljina_nosaca - duljina_prepusta * 2 & "m"
    LblPrepustLijevo.Text = duljina_prepusta & "m"
    LblPrepustDesno.Text = duljina_prepusta & "m"
    TrackBar1.Enabled = True
    TrackBar2.Enabled = True
    TrackBar3.Enabled = True
    TrackBar4.Enabled = True
End If
If nepomicni_oslonac_lijevo = True And pomicni_oslonac_desno = True And prepust_lijevo = True Then
    PictureBox9.Image = ImagemList1.Images(14)
    VelGrede.Text = duljina_nosaca - duljina_prepusta & "m"
    LblPrepustLijevo.Text = duljina_prepusta & "m"
    TrackBar1.Enabled = True

```

```

        TrackBar2.Enabled = True
        TrackBar3.Enabled = True
        TrackBar4.Enabled = True
    End If
    If nepomicni_oslonac_lijevo = True And pomicni_oslonac_desno = True And prepust_desno = True Then
        PictureBox9.Image = Imagelist1.Images(15)
        VelGrede.Text = duljina_nosaca - duljina_prepusta & "m"
        LblPrepustDesno.Text = duljina_prepusta & "m"
        TrackBar1.Enabled = True
        TrackBar2.Enabled = True
        TrackBar3.Enabled = True
        TrackBar4.Enabled = True
    End If
    If nepomicni_oslonac_lijevo = True And pomicni_oslonac_desno = True And prepust_obje_strane = True Then
        PictureBox9.Image = Imagelist1.Images(16)
        VelGrede.Text = duljina_nosaca - duljina_prepusta * 2 & "m"
        LblPrepustLijevo.Text = duljina_prepusta & "m"
        LblPrepustDesno.Text = duljina_prepusta & "m"
        TrackBar1.Enabled = True
        TrackBar2.Enabled = True
        TrackBar3.Enabled = True
        TrackBar4.Enabled = True
    End If
End If
If prepust_obje_strane = True Then
    razmak_oslonaca = duljina_nosaca - (2 * duljina_prepusta)
Else
    razmak_oslonaca = duljina_nosaca - duljina_prepusta
End If
sila1.kut = 90
sila2.kut = 90
sila3.kut = 90
sila4.kut = 90
If prepust_lijevo = True Then
    pozicijaA = duljina_prepusta
    pozicijaB = duljina_nosaca
ElseIf prepust_desno = True Then
    pozicijaA = 0
    pozicijaB = duljina_nosaca - duljina_prepusta
ElseIf prepust_obje_strane = True Then
    pozicijaA = duljina_prepusta
    pozicijaB = duljina_nosaca - duljina_prepusta
Else
    pozicijaA = 0
    pozicijaB = duljina_nosaca
End If
End Sub
Private Sub TrackBar1_Scroll(sender As Object, e As EventArgs) Handles TrackBar1.Scroll
    Label8.Text = TrackBar1.Value & "°"
    sila1.kut = TrackBar1.Value
End Sub
Private Sub TrackBar2_Scroll(sender As Object, e As EventArgs) Handles TrackBar2.Scroll
    Label14.Text = TrackBar2.Value & "°"
    sila2.kut = TrackBar2.Value
End Sub
Private Sub TrackBar3_Scroll(sender As Object, e As EventArgs) Handles TrackBar3.Scroll
    Label15.Text = TrackBar3.Value & "°"
    sila3.kut = TrackBar3.Value
End Sub
Private Sub TrackBar5_Scroll(sender As Object, e As EventArgs) Handles TrackBar4.Scroll

```

```

Label17.Text = TrackBar4.Value & "°"
sila4.kut = TrackBar4.Value
End Sub
Private Sub NumericUpDown1_ValueChanged_1(sender As Object, e As EventArgs) Handles NumericUpDown1.ValueChanged
    sila1.intenzitet = -NumericUpDown1.Value
End Sub
Private Sub NumericUpDown2_ValueChanged(sender As Object, e As EventArgs) Handles NumericUpDown2.ValueChanged
    NumericUpDown2.Maximum = duljina_nosaca
    sila1.pozicija = NumericUpDown2.Value
End Sub

Private Sub NumericUpDown3_ValueChanged(sender As Object, e As EventArgs) Handles NumericUpDown3.ValueChanged
    sila2.intenzitet = -NumericUpDown3.Value
End Sub
Private Sub NumericUpDown4_ValueChanged(sender As Object, e As EventArgs) Handles NumericUpDown4.ValueChanged
    NumericUpDown4.Maximum = duljina_nosaca
    sila2.pozicija = NumericUpDown4.Value
End Sub

Private Sub NumericUpDown8_ValueChanged(sender As Object, e As EventArgs) Handles NumericUpDown8.ValueChanged
    sila3.intenzitet = -NumericUpDown8.Value
End Sub
Private Sub NumericUpDown7_ValueChanged(sender As Object, e As EventArgs) Handles NumericUpDown7.ValueChanged
    NumericUpDown7.Maximum = duljina_nosaca
    sila3.pozicija = NumericUpDown7.Value
End Sub

Private Sub NumericUpDown6_ValueChanged(sender As Object, e As EventArgs) Handles NumericUpDown6.ValueChanged
    sila4.intenzitet = -NumericUpDown6.Value
End Sub
Private Sub NumericUpDown5_ValueChanged(sender As Object, e As EventArgs) Handles NumericUpDown5.ValueChanged
    NumericUpDown5.Maximum = duljina_nosaca
    sila4.pozicija = NumericUpDown5.Value
End Sub

Private Sub NumericUpDown10_ValueChanged(sender As Object, e As EventArgs) Handles NumericUpDown10.ValueChanged
    kontinuirano1.intenzitet = -NumericUpDown10.Value
    If NumericUpDown10.Value <> 0 Then
        NumericUpDown9.ForeColor = Color.Red
        NumericUpDown11.ForeColor = Color.Red
    End If
End Sub

Private Sub NumericUpDown9_ValueChanged(sender As Object, e As EventArgs) Handles NumericUpDown9.ValueChanged
    NumericUpDown9.Maximum = duljina_nosaca - 0.1
    kontinuirano1.pozicija_od = NumericUpDown9.Value
    If NumericUpDown9.Value > kontinuirano1.pozicija_do Then
        NumericUpDown9.ForeColor = Color.Red
        NumericUpDown11.ForeColor = Color.Red
    End If
End Sub

Private Sub NumericUpDown11_ValueChanged(sender As Object, e As EventArgs) Handles NumericUpDown11.ValueChanged
    NumericUpDown11.Minimum = kontinuirano1.pozicija_od + 0.1
    NumericUpDown11.Maximum = duljina_nosaca
    kontinuirano1.pozicija_do = NumericUpDown11.Value
    If NumericUpDown11.Value > kontinuirano1.pozicija_od Then
        NumericUpDown9.ForeColor = Color.Green
        NumericUpDown11.ForeColor = Color.Green
    End If
End Sub

Private Sub NumericUpDown12_ValueChanged(sender As Object, e As EventArgs) Handles NumericUpDown12.ValueChanged
    moment1.intenzitet = NumericUpDown12.Value
    If NumericUpDown12.Value <> 0 Then

```



```

        DomainUpDown1.ForeColor = Color.Red
    If DomainUpDown1.Text = "Smjer kazaljke na satu" Then
        DomainUpDown1.ForeColor = Color.Green
    ElseIf DomainUpDown1.Text = "Smjer obrnut od kazaljke na satu" Then
        DomainUpDown1.ForeColor = Color.Green
    End If
End If
End Sub
Private Sub NumericUpDown13_ValueChanged(sender As Object, e As EventArgs) Handles NumericUpDown13.ValueChanged
    NumericUpDown13.Maximum = duljina_nosaca
    moment1.pozicija = NumericUpDown13.Value
End Sub
Private Sub DomainUpDown1_SelectedItemChanged(sender As Object, e As EventArgs) Handles DomainUpDown1.SelectedItemChanged
    If DomainUpDown1.Text = "Smjer kazaljke na satu" Then
        moment1.smjer_kazaljke_na_satu = True
        moment1.intenzitet = (NumericUpDown12.Value)
        DomainUpDown1.ForeColor = Color.Green
    End If
    If DomainUpDown1.Text = "Smjer obrnut od kazaljke na satu" Then
        moment1.smjer_kazaljke_na_satu = False
        moment1.intenzitet = (NumericUpDown12.Value)
        DomainUpDown1.ForeColor = Color.Green
    End If
End Sub
Private Sub NumericUpDown15_ValueChanged(sender As Object, e As EventArgs) Handles NumericUpDown15.ValueChanged
    moment2.intenzitet = NumericUpDown15.Value
    If NumericUpDown12.Value <> 0 Then
        DomainUpDown2.ForeColor = Color.Red
        If DomainUpDown2.Text = "Smjer kazaljke na satu" Then
            DomainUpDown2.ForeColor = Color.Green
        ElseIf DomainUpDown2.Text = "Smjer obrnut od kazaljke na satu" Then
            DomainUpDown2.ForeColor = Color.Green
        End If
    End If
End Sub
Private Sub NumericUpDown14_ValueChanged(sender As Object, e As EventArgs) Handles NumericUpDown14.ValueChanged
    NumericUpDown14.Maximum = duljina_nosaca
    moment2.pozicija = NumericUpDown14.Value
End Sub
Private Sub DomainUpDown2_SelectedItemChanged(sender As Object, e As EventArgs) Handles DomainUpDown2.SelectedItemChanged
    If DomainUpDown2.Text = "Smjer kazaljke na satu" Then
        moment2.smjer_kazaljke_na_satu = True
        moment2.intenzitet = NumericUpDown15.Value
        DomainUpDown2.ForeColor = Color.Green
    End If
    If DomainUpDown2.Text = "Smjer obrnut od kazaljke na satu" Then
        moment2.smjer_kazaljke_na_satu = False
        moment1.intenzitet = NumericUpDown12.Value
        DomainUpDown2.ForeColor = Color.Green
    End If
End Sub
Private Sub Button26_Click(sender As Object, e As EventArgs) Handles Button26.Click
    TabControl11.SelectedIndex = TabControl11.SelectedIndex + 1 'klikom na gumb automatski prebacije na iduci tab (Pozicioniranje Sila)
    sila1.kut = (sila1.kut / 180) * System.Math.PI
    sila2.kut = (sila2.kut / 180) * System.Math.PI
    sila3.kut = (sila3.kut / 180) * System.Math.PI
    sila4.kut = (sila4.kut / 180) * System.Math.PI
    If kontinuirano1.intenzitet <> 0 Then
        vel_kont = kontinuirano1.pozicija_do - kontinuirano1.pozicija_od
    End If
End Sub

```

```

If DomainUpDown1.Text = "Smjer kazaljke na satu" Then
    moment1.intenzitet = -(moment1.intenzitet)
ElseIf DomainUpDown1.Text = "Smjer obrnut od kazaljke na satu" Then
    moment1.intenzitet = (moment1.intenzitet)
End If
If DomainUpDown2.Text = "Smjer kazaljke na satu" Then
    moment2.intenzitet = -(moment2.intenzitet)
ElseIf DomainUpDown2.Text = "Smjer obrnut od kazaljke na satu" Then
    moment2.intenzitet = (moment2.intenzitet)
End If
'nepomicni pomicni alg
If nepomicni_oslonac_lijevano = True And pomicni_oslonac_desno = True Then
    reakcijaAx = -(sila1.intenzitet * Math.Cos(sila1.kut) + sila2.intenzitet * Math.Cos(sila2.kut) + sila3.intenzitet *
Math.Cos(sila3.kut) + sila4.intenzitet * Math.Cos(sila4.kut))
    reakcijaBy = -((sila1.intenzitet * Math.Sin(sila1.kut) * (sila1.pozicija - pozicijaA) + sila2.intenzitet * Math.Sin(sila2.kut)
* (sila2.pozicija - pozicijaA) + sila3.intenzitet * Math.Sin(sila3.kut) * (sila3.pozicija - pozicijaA) + sila4.intenzitet *
Math.Sin(sila4.kut) * (sila4.pozicija - pozicijaA) + moment1.intenzitet + moment2.intenzitet + kontinuirano1.intenzitet * vel_kont *
((vel_kont / 2 + kontinuirano1.pozicija_od) - pozicijaA)) / razmak_oslonaca)
    reakcijaAy = -(sila1.intenzitet * Math.Sin(sila1.kut) + sila2.intenzitet * Math.Sin(sila2.kut) + sila3.intenzitet *
Math.Sin(sila3.kut) + sila4.intenzitet * Math.Sin(sila4.kut) + kontinuirano1.intenzitet * vel_kont + reakcijaBy)
End If
'pomicni nepomicni alg
If pomicni_oslonac_lijevano = True And nepomicni_oslonac_desno = True Then
    reakcijaBx = -(sila1.intenzitet * Math.Cos(sila1.kut) + sila2.intenzitet * Math.Cos(sila2.kut) + sila3.intenzitet *
Math.Cos(sila3.kut) + sila4.intenzitet * Math.Cos(sila4.kut))
    reakcijaAy = ((sila1.intenzitet * Math.Sin(sila1.kut) * (sila1.pozicija - pozicijaB) + sila2.intenzitet * Math.Sin(sila2.kut)
* (sila2.pozicija - pozicijaB) + sila3.intenzitet * Math.Sin(sila3.kut) * (sila3.pozicija - pozicijaB) + sila4.intenzitet *
Math.Sin(sila4.kut) * (sila4.pozicija - pozicijaB) + moment1.intenzitet + moment2.intenzitet + kontinuirano1.intenzitet * vel_kont *
((vel_kont / 2 + kontinuirano1.pozicija_od) - pozicijaB)) / razmak_oslonaca)
    reakcijaBy = (-sila1.intenzitet * Math.Sin(sila1.kut) - sila2.intenzitet * Math.Sin(sila2.kut) - sila3.intenzitet *
Math.Sin(sila3.kut) - sila4.intenzitet * Math.Sin(sila4.kut) - kontinuirano1.intenzitet * vel_kont - reakcijaAy)
End If
'pomicni pomicni alg
If pomicni_oslonac_lijevano = True And pomicni_oslonac_desno = True Then
    reakcijaAy = ((sila1.intenzitet * (sila1.pozicija - pozicijaB) + sila2.intenzitet * (sila2.pozicija - pozicijaB) +
sila3.intenzitet * (sila3.pozicija - pozicijaB) + sila4.intenzitet * (sila4.pozicija - pozicijaB) + moment1.intenzitet +
moment2.intenzitet + kontinuirano1.intenzitet * vel_kont * ((vel_kont / 2 + kontinuirano1.pozicija_od) - pozicijaB)) / razmak_oslonaca)
    reakcijaBy = (-sila1.intenzitet - sila2.intenzitet - sila3.intenzitet - sila4.intenzitet - kontinuirano1.intenzitet * vel_kont
- reakcijaAy)
End If
'ukljestenje
If ukljestenje = True Then
    reakcijaAx = -(sila1.intenzitet * Math.Cos(sila1.kut) + sila2.intenzitet * Math.Cos(sila2.kut) + sila3.intenzitet *
Math.Cos(sila3.kut) + sila4.intenzitet * Math.Cos(sila4.kut))
    reakcijaAy = -(sila1.intenzitet * Math.Sin(sila1.kut) + sila2.intenzitet * Math.Sin(sila2.kut) + sila3.intenzitet *
Math.Sin(sila3.kut) + sila4.intenzitet * Math.Sin(sila4.kut) + kontinuirano1.intenzitet * vel_kont)
    momentA = -(sila1.intenzitet * Math.Sin(sila1.kut) * sila1.pozicija + sila2.intenzitet * Math.Sin(sila2.kut) * sila2.pozicija
+ sila3.intenzitet * Math.Sin(sila3.kut) * sila3.pozicija + sila4.intenzitet * Math.Sin(sila4.kut) * sila4.pozicija +
kontinuirano1.intenzitet * vel_kont * (vel_kont / 2 + kontinuirano1.pozicija_od) + moment1.intenzitet + moment2.intenzitet)
End If

If reakcijaAx < 0.01 And reakcijaAx > -0.01 Then
    Label35.Text = "Nema reakcije"
Else Label35.Text = (reakcijaAx.ToString("0.000") & " [kN]")
End If

Label36.Text = (reakcijaAy.ToString("0.00") & " [kN]")

If reakcijaBx < 0.01 And reakcijaBx > -0.01 Then
    Label37.Text = "Nema reakcije"

```

```

Else Label137.Text = (reakcijaBx.ToString("0.000") & " [kN]")
End If

Label138.Text = (reakcijaBy.ToString("0.000") & " [kN]")

If ukljestenje = False Then
    Label140.Text = "Nema ukljestenja"
Else Label140.Text = (momentA.ToString("0.0") & " [kNm]")
End If

moments(0).pozicija = 0
moments(0).intenzitet = 0

If moment1.intenzitet <> 0 Then
    pomocna = 0
    If sila1.pozicija < moment1.pozicija Then
        pomocna = pomocna + sila1.intenzitet * Math.Sin(sila1.kut) * (moment1.pozicija - sila1.pozicija)
    End If
    If sila2.pozicija < moment1.pozicija Then
        pomocna = pomocna + sila2.intenzitet * Math.Sin(sila2.kut) * (moment1.pozicija - sila2.pozicija)
    End If
    If sila3.pozicija < moment1.pozicija Then
        pomocna = pomocna + sila3.intenzitet * Math.Sin(sila3.kut) * (moment1.pozicija - sila3.pozicija)
    End If
    If sila4.pozicija < moment1.pozicija Then
        pomocna = pomocna + sila4.intenzitet * Math.Sin(sila4.kut) * (moment1.pozicija - sila4.pozicija)
    End If
    If moment2.intenzitet <> 0 And moment2.pozicija < moment1.pozicija Then
        pomocna = pomocna - moment2.intenzitet
    End If
    If momentA <> 0 Then
        pomocna = pomocna - momentA
    End If
    If pozicijaA < moment1.pozicija Then
        pomocna = pomocna + reakcijaAy * (moment1.pozicija - pozicijaA)
    End If
    If pozicijaB < moment1.pozicija Then
        pomocna = pomocna + reakcijaBy * (moment1.pozicija - pozicijaB)
    End If
    If kontinuirano1.intenzitet <> 0 And kontinuirano1.pozicija_od < moment1.pozicija Then
        If kontinuirano1.pozicija_do <= moment1.pozicija Then
            pomocna = pomocna + kontinuirano1.intenzitet * (vel_kont) * ((vel_kont / 2) + (moment1.pozicija - kontinuirano1.pozicija_do))
        ElseIf kontinuirano1.pozicija_do > moment1.pozicija Then
            pomocna = pomocna + kontinuirano1.intenzitet * (moment1.pozicija - kontinuirano1.pozicija_od) * ((moment1.pozicija - kontinuirano1.pozicija_od) / 2)
        End If
    End If
    M_fleksijeM1 = pomocna
End If
moments(1).pozicija = moment1.pozicija
moments(1).intenzitet = M_fleksijeM1
''računanje momenata fleksije momenta2
If moment2.intenzitet <> 0 Then
    pomocna = 0
    If sila1.pozicija < moment2.pozicija Then
        pomocna = pomocna + sila1.intenzitet * Math.Sin(sila1.kut) * (moment2.pozicija - sila1.pozicija)
    End If
    If sila2.pozicija < moment2.pozicija Then
        pomocna = pomocna + sila2.intenzitet * Math.Sin(sila2.kut) * (moment2.pozicija - sila2.pozicija)
    End If

```

```

If sila3.pozicija < moment2.pozicija Then
    pomocna = pomocna + sila3.intenzitet * Math.Sin(sila3.kut) * (moment2.pozicija - sila3.pozicija)
End If
If sila4.pozicija < moment2.pozicija Then
    pomocna = pomocna + sila4.intenzitet * Math.Sin(sila4.kut) * (moment2.pozicija - sila4.pozicija)
End If
If moment1.intenzitet <> 0 And moment1.pozicija < moment2.pozicija Then
    pomocna = pomocna - moment1.intenzitet
End If
If momentA <> 0 Then
    pomocna = pomocna - momentA
End If
If pozicijaA < moment2.pozicija Then
    pomocna = pomocna + reakcijaAy * (moment2.pozicija - pozicijaA)
End If
If pozicijaB < moment2.pozicija Then
    pomocna = pomocna + reakcijaBy * (moment2.pozicija - pozicijaB)
End If
If kontinuirano1.intenzitet <> 0 And kontinuirano1.pozicija_od < moment2.pozicija Then
    If kontinuirano1.pozicija_do <= moment2.pozicija Then
        pomocna = pomocna + kontinuirano1.intenzitet * (vel_kont) * ((vel_kont / 2) + (moment2.pozicija -
kontinuirano1.pozicija_do))
    ElseIf kontinuirano1.pozicija_do > moment2.pozicija Then
        pomocna = pomocna + kontinuirano1.intenzitet * (moment2.pozicija - kontinuirano1.pozicija_od) * ((moment2.pozicija -
kontinuirano1.pozicija_od) / 2)
    End If
End If
M_fleksijeM2 = pomocna
End If
moments(2).pozicija = moment2.pozicija
moments(2).intenzitet = M_fleksijeM2

'racunanje momenata fleksije za silu 1
If sila1.intenzitet <> 0 Then
    pomocna = 0
    If sila2.pozicija <> 0 And sila2.pozicija <= sila1.pozicija Then
        pomocna = pomocna + sila2.intenzitet * Math.Sin(sila2.kut) * (sila1.pozicija - sila2.pozicija)
    End If
    If sila3.pozicija <> 0 And sila3.pozicija <= sila1.pozicija Then
        pomocna = pomocna + sila3.intenzitet * Math.Sin(sila3.kut) * (sila1.pozicija - sila3.pozicija)
    End If
    If sila4.pozicija <> 0 And sila4.pozicija <= sila1.pozicija Then
        pomocna = pomocna + sila4.intenzitet * Math.Sin(sila4.kut) * (sila1.pozicija - sila4.pozicija)
    End If
    If kontinuirano1.intenzitet <> 0 And kontinuirano1.pozicija_od < sila1.pozicija Then
        If kontinuirano1.pozicija_do < sila1.pozicija Then
            pomocna = pomocna + kontinuirano1.intenzitet * (vel_kont) * ((vel_kont / 2) + (sila1.pozicija -
kontinuirano1.pozicija_do))
        ElseIf kontinuirano1.pozicija_do >= sila1.pozicija Then
            pomocna = pomocna + kontinuirano1.intenzitet * (sila1.pozicija - kontinuirano1.pozicija_od) * ((sila1.pozicija -
kontinuirano1.pozicija_od) / 2)
        End If
    End If
    If moment1.intenzitet <> 0 And moment1.pozicija <= sila1.pozicija Then
        pomocna = pomocna - moment1.intenzitet
    End If
    If moment2.intenzitet <> 0 And moment2.pozicija <= sila1.pozicija Then
        pomocna = pomocna - moment2.intenzitet
    End If
    If momentA <> 0 Then

```

```

        pomocna = pomocna - momentA
    End If
    If pozicijaA <= sila1.pozicija Then
        pomocna = pomocna + reakcijaAy * (sila1.pozicija - pozicijaA)
    End If
    If pozicijaB <= sila1.pozicija Then
        pomocna = pomocna + reakcijaBy * (sila1.pozicija - pozicijaB)
    End If
    M_fleksijeS1 = pomocna
Else M_fleksijeS1 = 0
End If
moments(3).pozicija = sila1.pozicija
moments(3).intenzitet = M_fleksijeS1
'moment felkesije za silu 2
If sila2.intenzitet <> 0 Then
    pomocna = 0
    If sila1.intenzitet <> 0 And sila1.pozicija <= sila2.pozicija Then
        pomocna = pomocna + sila1.intenzitet * Math.Sin(sila1.kut) * (sila2.pozicija - sila1.pozicija)
    End If
    If sila3.intenzitet <> 0 And sila3.pozicija <= sila2.pozicija Then
        pomocna = pomocna + sila3.intenzitet * Math.Sin(sila3.kut) * (sila2.pozicija - sila3.pozicija)
    End If
    If sila4.intenzitet <> 0 And sila4.pozicija <= sila2.pozicija Then
        pomocna = pomocna + sila4.intenzitet * Math.Sin(sila4.kut) * (sila2.pozicija - sila4.pozicija)
    End If
    If kontinuirano1.intenzitet <> 0 And kontinuirano1.pozicija_od < sila2.pozicija Then
        If kontinuirano1.pozicija_do < sila2.pozicija Then
            pomocna = pomocna + kontinuirano1.intenzitet * (vel_kont) * ((vel_kont / 2) + (sila2.pozicija - kontinuirano1.pozicija_do))
        ElseIf kontinuirano1.pozicija_do >= sila2.pozicija Then
            pomocna = pomocna + kontinuirano1.intenzitet * (sila2.pozicija - kontinuirano1.pozicija_od) * ((sila2.pozicija - kontinuirano1.pozicija_od) / 2)
        End If
    End If
    If moment1.intenzitet <> 0 And moment1.pozicija <= sila2.pozicija Then
        pomocna = pomocna - moment1.intenzitet
    End If
    If moment2.intenzitet <> 0 And moment2.pozicija <= sila2.pozicija Then
        pomocna = pomocna - moment2.intenzitet
    End If
    If momentA <> 0 Then
        pomocna = pomocna - momentA
    End If
    If pozicijaA <= sila2.pozicija Then
        pomocna = pomocna + reakcijaAy * (sila2.pozicija - pozicijaA)
    End If
    If pozicijaB <= sila2.pozicija Then
        pomocna = pomocna + reakcijaBy * (sila2.pozicija - pozicijaB)
    End If
    M_fleksijeS2 = pomocna
Else M_fleksijeS2 = 0
End If
moments(4).pozicija = sila2.pozicija
moments(4).intenzitet = M_fleksijeS2
'racunanje momenata fleksije za silu 3
If sila3.intenzitet <> 0 Then
    pomocna = 0
    If sila2.intenzitet <> 0 And sila2.pozicija < sila3.pozicija Then
        pomocna = pomocna + sila2.intenzitet * Math.Sin(sila2.kut) * (sila3.pozicija - sila2.pozicija)
    End If

```

```

If sila1.intenzitet <> 0 And sila1.pozicija < sila3.pozicija Then
    pomocna = pomocna + sila1.intenzitet * Math.Sin(sila1.kut) * (sila3.pozicija - sila1.pozicija)
End If
If sila4.intenzitet <> 0 And sila4.pozicija < sila3.pozicija Then
    pomocna = pomocna + sila4.intenzitet * Math.Sin(sila4.kut) * (sila3.pozicija - sila4.pozicija)
End If
If kontinuirano1.intenzitet <> 0 And kontinuirano1.pozicija_od < sila3.pozicija Then
    If kontinuirano1.pozicija_do < sila3.pozicija Then
        pomocna = pomocna + kontinuirano1.intenzitet * (vel_kont) * ((vel_kont / 2) + (sila3.pozicija -
kontinuirano1.pozicija_do))
    ElseIf kontinuirano1.pozicija_do >= sila3.pozicija Then
        pomocna = pomocna + kontinuirano1.intenzitet * (sila3.pozicija - kontinuirano1.pozicija_od) * ((sila3.pozicija -
kontinuirano1.pozicija_od) / 2)
    End If
End If
If moment1.intenzitet <> 0 And moment1.pozicija <= sila3.pozicija Then
    pomocna = pomocna - moment1.intenzitet
End If
If moment2.intenzitet <> 0 And moment2.pozicija <= sila3.pozicija Then
    pomocna = pomocna - moment2.intenzitet
End If
If momentA <> 0 Then
    pomocna = pomocna - momentA
End If
If pozicijaA < sila3.pozicija Then
    pomocna = pomocna + reakcijaAy * (sila3.pozicija - pozicijaA)
End If
If pozicijaB < sila3.pozicija Then
    pomocna = pomocna + reakcijaBy * (sila3.pozicija - pozicijaB)
End If
M_fleksijeS3 = pomocna
Else M_fleksijeS3 = 0
End If
moments(5).pozicija = sila3.pozicija
moments(5).intenzitet = M_fleksijeS3
'racunanje momenata fleksije za silu 4
If sila4.intenzitet <> 0 Then
    pomocna = 0
    If sila2.intenzitet <> 0 And sila2.pozicija < sila4.pozicija Then
        pomocna = pomocna + sila2.intenzitet * Math.Sin(sila2.kut) * (sila4.pozicija - sila2.pozicija)
    End If
    If sila1.intenzitet <> 0 And sila1.pozicija < sila4.pozicija Then
        pomocna = pomocna + sila1.intenzitet * Math.Sin(sila1.kut) * (sila4.pozicija - sila1.pozicija)
    End If
    If sila3.intenzitet <> 0 And sila3.pozicija < sila4.pozicija Then
        pomocna = pomocna + sila3.intenzitet * Math.Sin(sila3.kut) * (sila4.pozicija - sila3.pozicija)
    End If
    If kontinuirano1.intenzitet <> 0 And kontinuirano1.pozicija_od < sila4.pozicija Then
        If kontinuirano1.pozicija_do < sila4.pozicija Then
            pomocna = pomocna + kontinuirano1.intenzitet * (vel_kont) * ((vel_kont / 2) + (sila4.pozicija -
kontinuirano1.pozicija_do))
        ElseIf kontinuirano1.pozicija_do >= sila4.pozicija Then
            pomocna = pomocna + kontinuirano1.intenzitet * (sila4.pozicija - kontinuirano1.pozicija_od) * ((sila4.pozicija -
kontinuirano1.pozicija_od) / 2)
        End If
    End If
    If moment1.intenzitet <> 0 And moment1.pozicija <= sila4.pozicija Then
        pomocna = pomocna - moment1.intenzitet
    End If
    If moment2.intenzitet <> 0 And moment2.pozicija <= sila4.pozicija Then

```

```

    pomocna = pomocna - moment2.intenzitet
End If
If momentA <> 0 Then
    pomocna = pomocna - momentA
End If
If pozicijaA < sila4.pozicija Then
    pomocna = pomocna + reakcijaAy * (sila4.pozicija - pozicijaA)
End If
If pozicijaB < sila4.pozicija Then
    pomocna = pomocna + reakcijaBy * (sila4.pozicija - pozicijaB)
End If
M_fleksijeS4 = pomocna
Else M_fleksijeS4 = 0
End If
moments(6).pozicija = sila4.pozicija
moments(6).intenzitet = M_fleksijeS4
''računanje momenata flekesije početka kontinuiranog
If kontinuirano1.intenzitet <> 0 Then
    pomocna = 0
    If kontinuirano1.pozicija_od > 0 Then
        If sila1.pozicija < kontinuirano1.pozicija_od Then
            pomocna = pomocna + sila1.intenzitet * Math.Sin(sila1.kut) * (kontinuirano1.pozicija_od - sila1.pozicija)
        End If
        If sila2.pozicija < kontinuirano1.pozicija_od Then
            pomocna = pomocna + sila2.intenzitet * Math.Sin(sila2.kut) * (kontinuirano1.pozicija_od - sila2.pozicija)
        End If
        If sila3.pozicija < kontinuirano1.pozicija_od Then
            pomocna = pomocna + sila3.intenzitet * Math.Sin(sila3.kut) * (kontinuirano1.pozicija_od - sila3.pozicija)
        End If
        If sila4.pozicija < kontinuirano1.pozicija_od Then
            pomocna = pomocna + sila4.intenzitet * Math.Sin(sila4.kut) * (kontinuirano1.pozicija_od - sila4.pozicija)
        End If
        If moment1.intenzitet <> 0 And moment1.pozicija < kontinuirano1.pozicija_od Then
            pomocna = pomocna - moment1.intenzitet
        End If
        If moment2.intenzitet <> 0 And moment2.pozicija < kontinuirano1.pozicija_od Then
            pomocna = pomocna - moment2.intenzitet
        End If
        If momentA <> 0 Then
            pomocna = pomocna - momentA
        End If
        If pozicijaA < kontinuirano1.pozicija_od Then
            pomocna = pomocna + reakcijaAy * (kontinuirano1.pozicija_od - pozicijaA)
        End If
        If pozicijaB < kontinuirano1.pozicija_od Then
            pomocna = pomocna + reakcijaBy * (kontinuirano1.pozicija_od - pozicijaB)
        End If
    End If
    M_fleksijeK1 = pomocna
End If
moments(7).pozicija = kontinuirano1.pozicija_od
moments(7).intenzitet = M_fleksijeK1
''računanje momenata flekesije kraja kontinuiranog
If kontinuirano1.intenzitet <> 0 Then
    pomocna = 0
    If kontinuirano1.pozicija_do < duljina_nosaca Then
        If sila1.pozicija > kontinuirano1.pozicija_do Then
            pomocna = pomocna + sila1.intenzitet * Math.Sin(sila1.kut) * (sila1.pozicija - kontinuirano1.pozicija_do)
        End If
        If sila2.pozicija > kontinuirano1.pozicija_do Then

```

```

        pomocna = pomocna + sila2.intenzitet * Math.Sin(sila2.kut) * (sila2.pozicija - kontinuirano1.pozicija_do)
    End If
    If sila3.pozicija > kontinuirano1.pozicija_do Then
        pomocna = pomocna + sila3.intenzitet * Math.Sin(sila3.kut) * (sila3.pozicija - kontinuirano1.pozicija_do)
    End If
    If sila4.pozicija > kontinuirano1.pozicija_do Then
        pomocna = pomocna + sila4.intenzitet * Math.Sin(sila4.kut) * (sila3.pozicija - kontinuirano1.pozicija_do)
    End If
    If moment1.intenzitet <> 0 And moment1.pozicija > kontinuirano1.pozicija_do Then
        pomocna = pomocna - moment1.intenzitet
    End If
    If moment2.intenzitet <> 0 And moment2.pozicija > kontinuirano1.pozicija_do Then
        pomocna = pomocna - moment2.intenzitet
    End If

    If pozicijaA > kontinuirano1.pozicija_do Then
        pomocna = pomocna + reakcijaAy * (pozicijaA - kontinuirano1.pozicija_do)
    End If
    If pozicijaB > kontinuirano1.pozicija_do Then
        pomocna = pomocna + reakcijaBy * (pozicijaB - kontinuirano1.pozicija_do)
    End If

    End If
    M_fleksijeK11 = pomocna
End If
moments(8).pozicija = kontinuirano1.pozicija_do
moments(8).intenzitet = M_fleksijeK11

'racunanje momenata felksije za oslonac B
pomocna = 0
If prepust_desno = False And prepust_obje_strane = False Then
    M_fleksijeB = 0
ElseIf prepust_desno = True Or prepust_obje_strane = True Then
    If sila1.intenzitet <> 0 And sila1.pozicija <= pozicijaB Then
        pomocna = pomocna + sila1.intenzitet * Math.Sin(sila1.kut) * (pozicijaB - sila1.pozicija)
    End If
    If sila2.intenzitet <> 0 And sila2.pozicija <= pozicijaB Then
        pomocna = pomocna + sila2.intenzitet * Math.Sin(sila2.kut) * (pozicijaB - sila2.pozicija)
    End If
    If sila3.intenzitet <> 0 And sila3.pozicija <= pozicijaB Then
        pomocna = pomocna + sila3.intenzitet * Math.Sin(sila3.kut) * (pozicijaB - sila3.pozicija)
    End If
    If sila4.intenzitet <> 0 And sila4.pozicija <= pozicijaB Then
        pomocna = pomocna + sila4.intenzitet * Math.Sin(sila4.kut) * (pozicijaB - sila4.pozicija)
    End If
    If kontinuirano1.intenzitet <> 0 And kontinuirano1.pozicija_od < pozicijaB Then
        If kontinuirano1.pozicija_do <= pozicijaB Then
            pomocna = pomocna + kontinuirano1.intenzitet * (vel_kont) * (((vel_kont) / 2) + (pozicijaB - kontinuirano1.pozicija_do))
        ElseIf kontinuirano1.pozicija_do >= pozicijaB Then
            pomocna = pomocna + kontinuirano1.intenzitet * (pozicijaB - kontinuirano1.pozicija_od) * ((pozicijaB - kontinuirano1.pozicija_od) / 2)
        End If
    End If
    If moment1.intenzitet <> 0 And moment1.pozicija <= pozicijaB Then
        pomocna = pomocna - moment1.intenzitet
    End If
    If moment2.intenzitet <> 0 And moment2.pozicija <= pozicijaB Then
        pomocna = pomocna - moment2.intenzitet
    End If

```



```

    If reakcijaAy <> 0 Then
        pomocna = pomocna + reakcijaAy * (pozicijaB - pozicijaA)
    End If
    M_fleksijeB = pomocna
    moments(9).pozicija = pozicijaB
    moments(9).intenzitet = M_fleksijeB
End If
'racunanje momenata felksije za oslonac A
pomocna = 0
If ukljesteenje = True Then
    M_fleksijeA = 0 - momentA
ElseIf prepust_lijevano = False And prepust_obje_strane = False Then
    M_fleksijeA = 0
ElseIf prepust_lijevano = True Or prepust_obje_strane = True Then
    If sila1.intenzitet <> 0 And sila1.pozicija <= duljina_prepusta Then
        pomocna = pomocna + sila1.intenzitet * Math.Sin(sila1.kut) * (duljina_prepusta - sila1.pozicija)
    End If
    If sila2.intenzitet <> 0 And sila2.pozicija <= duljina_prepusta Then
        pomocna = pomocna + sila2.intenzitet * Math.Sin(sila2.kut) * (duljina_prepusta - sila2.pozicija)
    End If
    If sila3.intenzitet <> 0 And sila3.pozicija <= duljina_prepusta Then
        pomocna = pomocna + sila3.intenzitet * Math.Sin(sila3.kut) * (duljina_prepusta - sila3.pozicija)
    End If
    If sila4.intenzitet <> 0 And sila4.pozicija <= duljina_prepusta Then
        pomocna = pomocna + sila4.intenzitet * Math.Sin(sila4.kut) * (duljina_prepusta - sila4.pozicija)
    End If
    If kontinuirano1.intenzitet <> 0 And kontinuirano1.pozicija_od < duljina_prepusta Then
        If kontinuirano1.pozicija_do <= duljina_prepusta Then
            pomocna = pomocna + kontinuirano1.intenzitet * (kontinuirano1.pozicija_do - kontinuirano1.pozicija_od) *
(((kontinuirano1.pozicija_do - kontinuirano1.pozicija_od) / 2) + duljina_prepusta - kontinuirano1.pozicija_do)
        ElseIf kontinuirano1.pozicija_do >= duljina_prepusta Then
            pomocna = pomocna + kontinuirano1.intenzitet * (duljina_prepusta - kontinuirano1.pozicija_od) * ((duljina_prepusta -
kontinuirano1.pozicija_od) / 2)
        End If
    End If
    If moment1.intenzitet <> 0 And moment1.pozicija <= duljina_prepusta Then
        pomocna = pomocna - moment1.intenzitet
    End If
    If moment2.intenzitet <> 0 And moment2.pozicija <= duljina_prepusta Then
        pomocna = pomocna - moment2.intenzitet
    End If
    M_fleksijeA = pomocna
End If
moments(10).pozicija = pozicijaA : moments(10).intenzitet = M_fleksijeA
moments(11).pozicija = duljina_nosaca
moments(11).intenzitet = 0
Call BubbleSort(moments)
Label52.Text = (M_fleksijeS1.ToString("0.000") & " [kNm]")
Label51.Text = (M_fleksijeS2.ToString("0.000") & " [kNm]")
Label50.Text = (M_fleksijeS3.ToString("0.000") & " [kNm]")
Label49.Text = (M_fleksijeS4.ToString("0.000") & " [kNm]")
Label54.Text = (M_fleksijeA.ToString("0.000") & " [kNm]")
Label53.Text = (M_fleksijeB.ToString("0.000") & " [kNm]")
Label48.Text = (M_fleksijeM1.ToString("0.000") & " [kNm]")
Label56.Text = (M_fleksijeM2.ToString("0.000") & " [kNm]")
Label59.Text = (M_fleksijeK1.ToString("0.000") & " [kNm]")
Label57.Text = (M_fleksijeK11.ToString("0.000") & " [kNm]")
Button27.PerformClick()
End Sub
Private Sub ResetButton_Click(sender As Object, e As EventArgs) Handles ResetButton.Click

```

```

Dim f2 As New Form1
f2.Show()
Me.Close()
End Sub
Private Sub Button27_Click(sender As Object, e As EventArgs) Handles Button27.Click
    Dim myGraphics As Graphics = GrafRezultati.CreateGraphics
    myGraphics.Clear(Color.White) 'čisti screen za crtanje
    Dim PenGreda As Pen
    Dim myFont As Font
    Dim pomocnaKut As Double
    myFont = New System.Drawing.Font("Segoe UI Emoji", 8, FontStyle.Bold)
    PenGreda = New Pen(Brushes.Black, 15)
    Dim pomocna As Single
    pomocna = duljina_nosaca
    Dim PenPomocna As Pen
    PenPomocna = New Pen(Brushes.Black, 0.5)
    Dim mjeriloX As Integer
    Dim mjeriloY As Single
    Dim avg As Double = (Math.Abs(reakcijaAy) + Math.Abs(reakcijaBy)) / 2
    If avg <= 20 Then
        mjeriloY = 10
    ElseIf avg > 20 And avg <= 40 Then
        mjeriloY = 5
    ElseIf avg > 40 And avg <= 80 Then
        mjeriloY = 2.5
    ElseIf avg > 80 And avg <= 160 Then
        mjeriloY = 1.25
    ElseIf avg > 160 And avg <= 320 Then
        mjeriloY = 0.625
    ElseIf avg > 320 And avg < 640 Then
        mjeriloY = 0.3125
    Else
        mjeriloY = 0.15625
    End If
    'CRTANJE NOSACA
    If pomocna <= 9 Then
        mjeriloX = 100
    ElseIf pomocna > 9 And pomocna <= 18 Then
        mjeriloX = 50
    ElseIf pomocna > 18 And pomocna <= 27 Then
        mjeriloX = 25
    End If
    pomocna = pomocna * mjeriloX
    myGraphics.DrawLine(PenGreda, 20, 40, pomocna + 20, 40)

    'CRTANJE OSOLONACA
    If ukljestenje = True Then
        myGraphics.DrawLine(PenGreda, 15, 0, 15, 80)
    Else
        pomocna = pozicijaA
        pomocna = (pomocna * mjeriloX) + 10
        If pomicni_oslonac_lijevo = True Then
            myGraphics.DrawImage(ImageList2.Images(0), pomocna, 50)
        Else
            myGraphics.DrawImage(ImageList2.Images(1), pomocna, 50)
        End If
        pomocna = pozicijaB
        If pomocna <= 9 Then
            pomocna = (pomocna * mjeriloX) + 10
            If pomicni_oslonac_desno = True Then

```

```

        myGraphics.DrawImage(ImageList2.Images(0), pomocna, 50)
    Else
        myGraphics.DrawImage(ImageList2.Images(1), pomocna, 50)
    End If
End If
End If
End If
'CRTANJE OPTERECENJA
Dim PenSila As Pen
PenSila = New Pen(Brushes.DarkBlue, 3.5)

Dim BrushBrojke As Brush
BrushBrojke = New Drawing.SolidBrush(Color.DarkBlue)
If sila1.intenzitet <> 0 Then
    pomocna = sila1.pozicija
    pomocna = (pomocna * mjeriloX) + 20
    myGraphics.DrawLine(PenSila, pomocna, 0, pomocna, 40)
    myGraphics.DrawLine(PenSila, pomocna, 40, pomocna + 10, 15)
    myGraphics.DrawLine(PenSila, pomocna, 40, pomocna - 10, 15)
    pomocnaKut = sila1.kut * (180 / Math.PI)
    myGraphics.DrawString("F1=" & sila1.intenzitet & "[kN]" & vbCrLf & "β1=" & pomocnaKut & "[°]", myFont, BrushBrojke, pomocna -
60, 0)

    myGraphics.DrawLine(PenPomocna, pomocna, 0, pomocna, 900)
End If
If sila2.intenzitet <> 0 Then
    pomocna = sila2.pozicija
    pomocna = (pomocna * mjeriloX) + 20
    myGraphics.DrawLine(PenSila, pomocna, 0, pomocna, 40)
    myGraphics.DrawLine(PenSila, pomocna, 40, pomocna + 10, 15)
    myGraphics.DrawLine(PenSila, pomocna, 40, pomocna - 10, 15)
    pomocnaKut = sila2.kut * (180 / Math.PI)
    myGraphics.DrawString("F2=" & sila2.intenzitet & "[kN]" & vbCrLf & "β2=" & pomocnaKut & "[°]", myFont, BrushBrojke, pomocna -
60, 0)

    myGraphics.DrawLine(PenPomocna, pomocna, 0, pomocna, 900)
End If
If sila3.intenzitet <> 0 Then
    pomocna = sila3.pozicija
    pomocna = (pomocna * mjeriloX) + 20
    myGraphics.DrawLine(PenSila, pomocna, 0, pomocna, 40)
    myGraphics.DrawLine(PenSila, pomocna, 40, pomocna + 10, 15)
    myGraphics.DrawLine(PenSila, pomocna, 40, pomocna - 10, 15)
    pomocnaKut = sila3.kut * (180 / Math.PI)
    myGraphics.DrawString("F3=" & sila3.intenzitet & "[kN]" & vbCrLf & "β3=" & pomocnaKut & "[°]", myFont, BrushBrojke, pomocna -
60, 0)

    myGraphics.DrawLine(PenPomocna, pomocna, 0, pomocna, 900)
End If
If sila4.intenzitet <> 0 Then
    pomocna = sila4.pozicija
    pomocna = (pomocna * mjeriloX) + 20
    myGraphics.DrawLine(PenSila, pomocna, 0, pomocna, 40)
    myGraphics.DrawLine(PenSila, pomocna, 40, pomocna + 10, 15)
    myGraphics.DrawLine(PenSila, pomocna, 40, pomocna - 10, 15)
    pomocnaKut = sila4.kut * (180 / Math.PI)
    myGraphics.DrawString("F4=" & sila4.intenzitet & "[kN]" & vbCrLf & "β4=" & pomocnaKut & "[°]", myFont, BrushBrojke, pomocna -
60, 0)

    myGraphics.DrawLine(PenPomocna, pomocna, 0, pomocna, 900)
End If
If kontinuirano1.intenzitet <> 0 Then
    BrushBrojke = New Drawing.SolidBrush(Color.Green)
    Dim penKont As Pen
    penKont = New Pen(Brushes.Green, 2.5)

```

```

pomocna = kontinuirano1.pozicija_od
pomocna = (pomocna * mjeriloX) + 20
myGraphics.DrawLine(PenPomocna, pomocna, 0, pomocna, 900)
Dim pomocna2 As Single
pomocna2 = kontinuirano1.pozicija_do
pomocna2 = (pomocna2 * mjeriloX) + 20
myGraphics.DrawLine(PenPomocna, pomocna2, 0, pomocna2, 900)
Dim points As Point() = {New Point(pomocna, 40), New Point(pomocna, 15), New Point(pomocna2, 15), New Point(pomocna2, 40)}
myGraphics.DrawLines(penKont, points)
myGraphics.DrawLine(penKont, pomocna, 40, pomocna + 5, 20)
myGraphics.DrawLine(penKont, pomocna, 40, pomocna - 5, 20)
myGraphics.DrawLine(penKont, pomocna2, 40, pomocna2 + 5, 20)
myGraphics.DrawLine(penKont, pomocna2, 40, pomocna2 - 5, 20)
myGraphics.DrawString("q=" & kontinuirano1.intenzitet & "[kN/m]", myFont, BrushBrojke, pomocna2 + 5, 15)
End If
If moment1.intenzitet <> 0 Then
    BrushBrojke = New Drawing.SolidBrush(Color.OrangeRed)
    pomocna = (moment1.pozicija * mjeriloX) + 10
    If moment1.smjer_kazaljke_na_satu Then
        myGraphics.DrawImage(ImageList2.Images(2), pomocna, 10)
        myGraphics.DrawString("M1=" & moment1.intenzitet & "[kNm]", myFont, BrushBrojke, pomocna + 20, 23)
    Else
        myGraphics.DrawImage(ImageList2.Images(3), pomocna, 10)
        myGraphics.DrawString("M1=" & moment1.intenzitet & "[kNm]", myFont, BrushBrojke, pomocna + 20, 23)
    End If
End If
If moment2.intenzitet <> 0 Then
    BrushBrojke = New Drawing.SolidBrush(Color.OrangeRed)
    pomocna = (moment2.pozicija * mjeriloX) + 10
    If moment2.smjer_kazaljke_na_satu Then
        myGraphics.DrawImage(ImageList2.Images(2), pomocna, 10)
        myGraphics.DrawString("M2=" & moment2.intenzitet & "[kNm]", myFont, BrushBrojke, pomocna + 20, 23)
    Else
        myGraphics.DrawImage(ImageList2.Images(3), pomocna, 10)
        myGraphics.DrawString("M2=" & moment2.intenzitet & "[kNm]", myFont, BrushBrojke, pomocna + 20, 23)
    End If
End If

End If
'CRTANJE REAKCIJSKIH SILA
PenSila = New Pen(Brushes.Red, 2.5)
BrushBrojke = New Drawing.SolidBrush(Color.Red)
If ukljestenje Then

    If momentA <> 0 Then
        myGraphics.DrawImage(ImageList2.Images(4), -5, 40)
        myGraphics.DrawString("M", myFont, BrushBrojke, 0, 58)
    End If
    If reakcijaAy <> 0 Then
        pomocna = 20
        myGraphics.DrawLine(PenSila, pomocna, 85, pomocna, 40)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna + 5, 60)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna - 5, 60)
        myGraphics.DrawString("FAy", myFont, BrushBrojke, pomocna + 5, 75)
        myGraphics.DrawLine(PenPomocna, pomocna, 0, pomocna, 900)
    End If
    If reakcijaAx <> 0 Then
        pomocna = 20
        myGraphics.DrawLine(PenSila, 0, 40, pomocna, 40)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna - 5, 45)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna - 5, 35)
    End If

```

```

        myGraphics.DrawString("FAx", myFont, BrushBrojke, 0, 23)
    End If
Else
    pomocna = (pozicijaA * mjeriloX) + 20

    If pomocni_oslonac_lijevo Then
        myGraphics.DrawLine(PenSila, pomocna, 85, pomocna, 40)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna + 5, 50)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna - 5, 50)
        myGraphics.DrawString("FAy", myFont, BrushBrojke, pomocna + 5, 75)
        myGraphics.DrawLine(PenPomocna, pomocna, 0, pomocna, 900)
    Else
        myGraphics.DrawLine(PenSila, pomocna, 85, pomocna, 40)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna + 5, 50)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna - 5, 50)
        myGraphics.DrawString("FAy", myFont, BrushBrojke, pomocna + 5, 75)
        myGraphics.DrawLine(PenSila, 0, 40, pomocna, 40)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna - 5, 45)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna - 5, 35)
        myGraphics.DrawString("FAx", myFont, BrushBrojke, 0, 23)
        myGraphics.DrawLine(PenPomocna, pomocna, 0, pomocna, 900)
    End If

    pomocna = (pozicijaB * mjeriloX) + 20
    If pomocni_oslonac_desno Then
        myGraphics.DrawLine(PenSila, pomocna, 85, pomocna, 40)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna + 5, 50)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna - 5, 50)
        myGraphics.DrawString("FBy", myFont, BrushBrojke, pomocna + 5, 75)
        myGraphics.DrawLine(PenPomocna, pomocna, 0, pomocna, 900)
    Else
        myGraphics.DrawLine(PenSila, pomocna, 85, pomocna, 40)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna + 5, 50)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna - 5, 50)
        myGraphics.DrawString("FBy", myFont, BrushBrojke, pomocna + 5, 75)
        myGraphics.DrawLine(PenSila, pomocna + 20, 40, pomocna, 40)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna + 5, 45)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna + 5, 35)
        myGraphics.DrawString("FBx", myFont, BrushBrojke, pomocna, 23)
        myGraphics.DrawLine(PenPomocna, pomocna, 0, pomocna, 900)
    End If
End If

'GRAF OPTERECENJA NA X-osi

Dim stopX As Single          'zadnja tocka x
Dim stopY As Single         'zadnja tocka y
Dim a As Single              'duljina nosaca
Dim s1 As Single             'pozicija sile 1
Dim s2 As Single             'pozicija sile2
Dim s3 As Single             'pozicija sile3
Dim s4 As Single             'pozicija sile4
Dim rA As Single             'pozicija reakcije a
Dim rB As Single             'pozicija reakcije b
Dim k1 As Single             'pozicija početka kontinuiranog1
Dim k11 As Single            'pozicija kraja kontinuiranog

Dim pomocna1 As Single
a = duljina_nosaca * mjeriloX
s1 = sila1.pozicija * mjeriloX
s2 = sila2.pozicija * mjeriloX
s3 = sila3.pozicija * mjeriloX

```

```

s4 = sila4.pozicija * mjeriloX
k1 = kontinuirano1.pozicija_od * mjeriloX
k11 = kontinuirano1.pozicija_do * mjeriloX
rA = pozicijaA * mjeriloX
rB = pozicijaB * mjeriloX

Dim PenGraf As Pen
PenGraf = New Pen(Brushes.Black, 9)
If reakcijaAx <> 0 Or reakcijaBx <> 0 Then
    pomocna = duljina_nosaca * mjeriloX + 20
    myGraphics.DrawLine(PenGraf, 20, 220, pomocna, 220) 'crtanje x osi grafa
    pomocniString = "[X]"
    myGraphics.DrawString(pomocniString, myFont, BrushBrojke, 0, 210)
    stopX = 20
    stopY = 220
    PenGraf = New Pen(Brushes.Blue, 3)
    Dim brojac As Single
    For brojac = 0 To a Step 1
        If brojac = rA And reakcijaAx <> 0 Then
            stopX = (brojac) + 20
            pomocna = stopY - (reakcijaAx * mjeriloY)
            myGraphics.DrawLine(PenGraf, stopX, stopY, stopX, pomocna)
            pomocniString = reakcijaAx.ToString("0.000")
            myGraphics.DrawString(pomocniString, myFont, BrushBrojke, stopX, pomocna)
            stopY = pomocna
        End If
        If brojac = rB And reakcijaBx <> 0 Then
            stopX = (brojac) + 20
            pomocna = stopY - (reakcijaBx * mjeriloY)
            myGraphics.DrawLine(PenGraf, stopX, stopY, stopX, pomocna)
            pomocniString = reakcijaBx.ToString("0.000")
            myGraphics.DrawString(pomocniString, myFont, BrushBrojke, stopX, pomocna)
            stopY = pomocna
        End If
        If brojac = s1 And sila1.intenzitet * Math.Cos(sila1.kut) <> 0 Then
            stopX = (brojac) + 20
            pomocna = stopY - ((sila1.intenzitet * Math.Cos(sila1.kut)) * mjeriloY)
            myGraphics.DrawLine(PenGraf, stopX, stopY, stopX, pomocna)
            pomocniString = (sila1.intenzitet * Math.Cos(sila1.kut)).ToString("0.000")
            myGraphics.DrawString(pomocniString, myFont, BrushBrojke, stopX, pomocna)
            stopY = pomocna
        End If
        If brojac = s2 And sila2.intenzitet * Math.Cos(sila2.kut) <> 0 Then
            stopX = (brojac) + 20
            pomocna = stopY - ((sila2.intenzitet * Math.Cos(sila2.kut)) * mjeriloY)
            myGraphics.DrawLine(PenGraf, stopX, stopY, stopX, pomocna)
            pomocniString = (sila2.intenzitet * Math.Cos(sila2.kut)).ToString("0.000")
            myGraphics.DrawString(pomocniString, myFont, BrushBrojke, stopX, pomocna)
            stopY = pomocna
        End If
        If brojac = s3 And sila3.intenzitet * Math.Cos(sila3.kut) <> 0 Then
            stopX = (brojac) + 20
            pomocna = stopY - ((sila3.intenzitet * Math.Cos(sila3.kut)) * mjeriloY)
            myGraphics.DrawLine(PenGraf, stopX, stopY, stopX, pomocna)
            pomocniString = (sila3.intenzitet * Math.Cos(sila3.kut)).ToString("0.000")
            myGraphics.DrawString(pomocniString, myFont, BrushBrojke, stopX, pomocna)
            stopY = pomocna
        End If
        If brojac = s4 And sila4.intenzitet * Math.Cos(sila4.kut) <> 0 Then
            stopX = (brojac) + 20

```

```

        pomocna = stopY - ((sila4.intenzitet * Math.Cos(sila4.kut)) * mjeriloY)
        myGraphics.DrawLine(PenGraf, stopX, stopY, stopX, pomocna)
        pomocniString = (sila4.intenzitet * Math.Cos(sila4.kut)).ToString("0.000")
        myGraphics.DrawString(pomocniString, myFont, BrushBrojke, stopX, pomocna)
        stopY = pomocna
    End If
    pomocna = brojac + 20
    myGraphics.DrawLine(PenGraf, stopX, stopY, pomocna, stopY)
Next
End If
'CRTANJE GRAFOVA ZA Y-os
PenGraf = New Pen(Brushes.Black, 9)
pomocna = duljina_nosaca * mjeriloX + 20
myGraphics.DrawLine(PenGraf, 20, 450, pomocna, 450) 'crtanje x osi grafa
pomocniString = "[Y]"
myGraphics.DrawString(pomocniString, myFont, BrushBrojke, 0, 440)
stopX = 20
stopY = 450
PenGraf = New Pen(Brushes.Blue, 3)
For brojac = 0 To a Step 1
    If brojac = rA And reakcijaAy <> 0 Then
        PenGraf = New Pen(Brushes.Blue, 3)
        stopX = (brojac) + 20
        pomocna = stopY - (reakcijaAy * mjeriloY)
        myGraphics.DrawLine(PenGraf, stopX, stopY, stopX, pomocna)
        pomocniString = reakcijaAy.ToString("0.000")
        myGraphics.DrawString(pomocniString, myFont, BrushBrojke, stopX, pomocna)
        stopY = pomocna
    End If
    If brojac = rB And reakcijaBy <> 0 Then
        PenGraf = New Pen(Brushes.Blue, 3)
        stopX = (brojac) + 20
        pomocna = stopY - (reakcijaBy * mjeriloY)
        myGraphics.DrawLine(PenGraf, stopX, stopY, stopX, pomocna)
        pomocniString = reakcijaBy.ToString("0.000")
        myGraphics.DrawString(pomocniString, myFont, BrushBrojke, stopX, pomocna)
        stopY = pomocna
    End If
    If brojac = s1 And sila1.intenzitet <> 0 Then
        PenGraf = New Pen(Brushes.Blue, 3)
        stopX = (brojac) + 20
        pomocna = stopY - ((sila1.intenzitet * Math.Sin(sila1.kut)) * mjeriloY)
        myGraphics.DrawLine(PenGraf, stopX, stopY, stopX, pomocna)
        pomocniString = (sila1.intenzitet * Math.Sin(sila1.kut)).ToString("0.000")
        myGraphics.DrawString(pomocniString, myFont, BrushBrojke, stopX, pomocna)
        stopY = pomocna
    End If
    If brojac = s2 And sila2.intenzitet <> 0 Then
        PenGraf = New Pen(Brushes.Blue, 3)
        stopX = (brojac) + 20
        pomocna = stopY - ((sila2.intenzitet * Math.Sin(sila2.kut)) * mjeriloY)
        myGraphics.DrawLine(PenGraf, stopX, stopY, stopX, pomocna)
        pomocniString = (sila2.intenzitet * Math.Sin(sila2.kut)).ToString("0.000")
        myGraphics.DrawString(pomocniString, myFont, BrushBrojke, stopX, pomocna)
        stopY = pomocna
    End If
    If brojac = s3 And sila3.intenzitet <> 0 Then
        PenGraf = New Pen(Brushes.Blue, 3)
        stopX = (brojac) + 20
        pomocna = stopY - ((sila3.intenzitet * Math.Sin(sila3.kut)) * mjeriloY)

```

```

myGraphics.DrawLine(PenGraf, stopX, stopY, stopX, pomocna)
pomocniString = (sila3.intenzitet * Math.Sin(sila3.kut)).ToString("0.000")
myGraphics.DrawString(pomocniString, myFont, BrushBrojke, stopX, pomocna)
stopY = pomocna
End If
If brojac = s4 And sila4.intenzitet <> 0 Then
    PenGraf = New Pen(Brushes.Blue, 3)
    stopX = (brojac) + 20
    pomocna = stopY - ((sila4.intenzitet * Math.Sin(sila4.kut)) * mjeriloY)
    myGraphics.DrawLine(PenGraf, stopX, stopY, stopX, pomocna)
    pomocniString = (sila4.intenzitet * Math.Sin(sila4.kut)).ToString("0.000")
    myGraphics.DrawString(pomocniString, myFont, BrushBrojke, stopX, pomocna)
    stopY = pomocna
End If
If brojac >= k1 And brojac <= k11 And kontinuirano1.intenzitet <> 0 Then
    pomocna = brojac + 20
    Dim D As Single = (mjeriloX / mjeriloY) / 10
    pomocna1 = stopY - (kontinuirano1.intenzitet / mjeriloY) / (mjeriloX / mjeriloY / mjeriloY)
    myGraphics.DrawLine(PenGraf, stopX, stopY, pomocna, pomocna1)
    stopX = pomocna
    stopY = pomocna1
    If pomocna1 = 450 And pomocna < a And pomocna > 0 And pomocna > k1 And pomocna < k11 Then
        mjestoMax = pomocna + 1
    End If
Else
    PenGraf = New Pen(Brushes.Blue, 3)
    pomocna = brojac + 20
    myGraphics.DrawLine(PenGraf, stopX, stopY, pomocna, stopY)
End If
Next

If mjestoMax <> 0 Then
    Dim temp As Double = (mjestoMax - 20) / mjeriloX
    pomocna = 0
    If sila1.intenzitet <> 0 And sila1.pozicija <= temp Then
        pomocna = pomocna + sila1.intenzitet * Math.Sin(sila1.kut) * (temp - sila1.pozicija)
    End If
    If sila2.intenzitet <> 0 And sila2.pozicija <= temp Then
        pomocna = pomocna + sila2.intenzitet * Math.Sin(sila2.kut) * (temp - sila2.pozicija)
    End If
    If sila3.intenzitet <> 0 And sila3.pozicija <= temp Then
        pomocna = pomocna + sila3.intenzitet * Math.Sin(sila3.kut) * (temp - sila3.pozicija)
    End If
    If sila4.intenzitet <> 0 And sila4.pozicija <= temp Then
        pomocna = pomocna + sila4.intenzitet * Math.Sin(sila4.kut) * (temp - sila4.pozicija)
    End If
    If kontinuirano1.intenzitet <> 0 And kontinuirano1.pozicija_od < temp Then
        If kontinuirano1.pozicija_do < temp Then
            pomocna = pomocna + kontinuirano1.intenzitet * (vel_kont) * ((vel_kont / 2) + (temp - kontinuirano1.pozicija_do))
        ElseIf kontinuirano1.pozicija_do > temp Then
            pomocna = pomocna + kontinuirano1.intenzitet * (temp - kontinuirano1.pozicija_od) * ((temp -
kontinuirano1.pozicija_od) / 2)
        End If
    End If
    If moment1.intenzitet <> 0 And moment1.pozicija <= temp Then
        pomocna = pomocna - moment1.intenzitet
    End If
    If moment2.intenzitet <> 0 And moment2.pozicija <= temp Then
        pomocna = pomocna - moment2.intenzitet
    End If

```



```

        If pozicijaA <= temp Then
            pomocna = pomocna + reakcijaAy * (temp - pozicijaA)
        End If
        If pozicijaB <= temp Then
            pomocna = pomocna + reakcijaBy * (temp - pozicijaB)
        End If
        iznosMax = pomocna
    Else iznosMax = 0
        mjestoMax = 0
    End If
End Sub
Private Sub Button28_Click(sender As Object, e As EventArgs) Handles Button28.Click
    Dim myGraphics As Graphics = GrafRezultati.CreateGraphics
    myGraphics.Clear(Color.White) ''čisti screen za crtanje
    Dim PenGreda As Pen
    Dim myFont As Font
    Dim pomocnaKut As Double
    myFont = New System.Drawing.Font("Segoe UI Emoji", 8, FontStyle.Bold)
    PenGreda = New Pen(Brushes.Black, 15)
    Dim pomocna As Single
    pomocna = duljina_nosaca
    Dim PenPomocna As Pen
    PenPomocna = New Pen(Brushes.Black, 0.5)
    Dim mjeriloX As Integer
    Dim mjeriloY As Single
    Dim i As Integer
    Dim najveci As Double = 0
    For i = 0 To 11
        If Math.Abs(moments(i).intenzitet) > najveci Then
            najveci = Math.Abs(moments(i).intenzitet)
        End If
    Next
    If najveci < iznosMax Then
        najveci = iznosMax
    End If
    If najveci > 0 Then
        mjeriloY = 10
    End If
    If najveci > 25 Then
        mjeriloY = 5
    End If
    If najveci > 50 Then
        mjeriloY = 2.5
    End If
    If najveci > 100 Then
        mjeriloY = 1.125
    End If
    If najveci > 200 Then
        mjeriloY = 0.5625
    End If
    If najveci > 400 Then
        mjeriloY = 0.28125
    End If
    'CRTANJE NOSACA
    If pomocna <= 9 Then
        mjeriloX = 100
    ElseIf pomocna > 9 And pomocna <= 18 Then
        mjeriloX = 50
    ElseIf pomocna > 18 And pomocna <= 27 Then
        mjeriloX = 25
    End If

```

```

End If
pomocna = pomocna * mjeriloX
myGraphics.DrawLine(PenGreda, 20, 40, pomocna + 20, 40)

'CRTANJE OSLONACA
If ukljestenje = True Then
    myGraphics.DrawLine(PenGreda, 15, 0, 15, 80)
Else
    pomocna = pozicijaA
    pomocna = (pomocna * mjeriloX) + 10
    If pomicni_oslonac_lijevo = True Then
        myGraphics.DrawImage(ImageList2.Images(0), pomocna, 50)
    Else
        myGraphics.DrawImage(ImageList2.Images(1), pomocna, 50)
    End If
    pomocna = pozicijaB
    If pomocna <= 9 Then
        pomocna = (pomocna * mjeriloX) + 10
        If pomicni_oslonac_desno = True Then
            myGraphics.DrawImage(ImageList2.Images(0), pomocna, 50)
        Else
            myGraphics.DrawImage(ImageList2.Images(1), pomocna, 50)
        End If
    End If
End If

'CRTANJE OPTERECENJA
Dim PenSila As Pen
PenSila = New Pen(Brushes.DarkBlue, 3.5)
Dim BrushBrojke As Brush
BrushBrojke = New Drawing.SolidBrush(Color.DarkBlue)
If moment1.intenzitet <> 0 Then
    BrushBrojke = New Drawing.SolidBrush(Color.OrangeRed)
    pomocna = (moment1.pozicija * mjeriloX) + 10
    myGraphics.DrawLine(PenPomocna, pomocna + 10, 0, pomocna + 10, 900)
    If moment1.smjer_kazaljke_na_satu Then
        myGraphics.DrawImage(ImageList2.Images(2), pomocna, 10)
        myGraphics.DrawString("M1=" & moment1.intenzitet & "[kNm]", myFont, BrushBrojke, pomocna + 20, 23)
    Else
        myGraphics.DrawImage(ImageList2.Images(3), pomocna, 10)
        myGraphics.DrawString("M1=" & moment1.intenzitet & "[kNm]", myFont, BrushBrojke, pomocna + 20, 23)
    End If
End If

If moment2.intenzitet <> 0 Then
    BrushBrojke = New Drawing.SolidBrush(Color.OrangeRed)
    pomocna = (moment2.pozicija * mjeriloX) + 10
    myGraphics.DrawLine(PenPomocna, pomocna + 10, 0, pomocna + 10, 900)
    If moment2.smjer_kazaljke_na_satu Then
        myGraphics.DrawImage(ImageList2.Images(2), pomocna, 10)
        myGraphics.DrawString("M2=" & moment2.intenzitet & "[kNm]", myFont, BrushBrojke, pomocna + 20, 23)
    Else
        myGraphics.DrawImage(ImageList2.Images(3), pomocna, 10)
        myGraphics.DrawString("M2=" & moment2.intenzitet & "[kNm]", myFont, BrushBrojke, pomocna + 20, 23)
    End If
End If

BrushBrojke = New Drawing.SolidBrush(Color.DarkBlue)
If sila1.intenzitet <> 0 Then
    pomocna = sila1.pozicija
    pomocna = (pomocna * mjeriloX) + 20
    myGraphics.DrawLine(PenSila, pomocna, 0, pomocna, 40)

```

```

myGraphics.DrawLine(PenSila, pomocna, 40, pomocna + 10, 15)
myGraphics.DrawLine(PenSila, pomocna, 40, pomocna - 10, 15)
pomocnaKut = sila1.kut * (180 / Math.PI)
myGraphics.DrawString("F1=" & sila1.intenzitet & "[kN]" & vbCrLf & "β1=" & pomocnaKut & "[°]", myFont, BrushBrojke, pomocna -
60, 0)

myGraphics.DrawLine(PenPomocna, pomocna, 0, pomocna, 900)
End If
If sila2.intenzitet <> 0 Then
pomocna = sila2.pozicija
pomocna = (pomocna * mjeriloX) + 20
myGraphics.DrawLine(PenSila, pomocna, 0, pomocna, 40)
myGraphics.DrawLine(PenSila, pomocna, 40, pomocna + 10, 15)
myGraphics.DrawLine(PenSila, pomocna, 40, pomocna - 10, 15)
pomocnaKut = sila2.kut * (180 / Math.PI)
myGraphics.DrawString("F2=" & sila2.intenzitet & "[kN]" & vbCrLf & "β2=" & pomocnaKut & "[°]", myFont, BrushBrojke, pomocna -
60, 0)

myGraphics.DrawLine(PenPomocna, pomocna, 0, pomocna, 900)
End If
If sila3.intenzitet <> 0 Then
pomocna = sila3.pozicija
pomocna = (pomocna * mjeriloX) + 20
myGraphics.DrawLine(PenSila, pomocna, 0, pomocna, 40)
myGraphics.DrawLine(PenSila, pomocna, 40, pomocna + 10, 15)
myGraphics.DrawLine(PenSila, pomocna, 40, pomocna - 10, 15)
pomocnaKut = sila3.kut * (180 / Math.PI)
myGraphics.DrawString("F3=" & sila3.intenzitet & "[kN]" & vbCrLf & "β3=" & pomocnaKut & "[°]", myFont, BrushBrojke, pomocna -
60, 0)

myGraphics.DrawLine(PenPomocna, pomocna, 0, pomocna, 900)
End If
If sila4.intenzitet <> 0 Then
pomocna = sila4.pozicija
pomocna = (pomocna * mjeriloX) + 20
myGraphics.DrawLine(PenSila, pomocna, 0, pomocna, 40)
myGraphics.DrawLine(PenSila, pomocna, 40, pomocna + 10, 15)
myGraphics.DrawLine(PenSila, pomocna, 40, pomocna - 10, 15)
pomocnaKut = sila4.kut * (180 / Math.PI)
myGraphics.DrawString("F4=" & sila4.intenzitet & "[kN]" & vbCrLf & "β4=" & pomocnaKut & "[°]", myFont, BrushBrojke, pomocna -
60, 0)

myGraphics.DrawLine(PenPomocna, pomocna, 0, pomocna, 900)
End If
If kontinuirano1.intenzitet <> 0 Then
BrushBrojke = New Drawing.SolidBrush(Color.Green)
Dim penKont As Pen
penKont = New Pen(Brushes.Green, 2.5)
pomocna = kontinuirano1.pozicija_od
pomocna = (pomocna * mjeriloX) + 20
myGraphics.DrawLine(PenPomocna, pomocna, 0, pomocna, 900)
Dim pomocna2 As Single
pomocna2 = kontinuirano1.pozicija_do
pomocna2 = (pomocna2 * mjeriloX) + 20
myGraphics.DrawLine(PenPomocna, pomocna2, 0, pomocna2, 900)
Dim points As Point() = {New Point(pomocna, 40), New Point(pomocna, 15), New Point(pomocna2, 15), New Point(pomocna2, 40)}
myGraphics.DrawLines(penKont, points)
myGraphics.DrawLine(penKont, pomocna, 40, pomocna + 5, 20)
myGraphics.DrawLine(penKont, pomocna, 40, pomocna - 5, 20)
myGraphics.DrawLine(penKont, pomocna2, 40, pomocna2 + 5, 20)
myGraphics.DrawLine(penKont, pomocna2, 40, pomocna2 - 5, 20)
myGraphics.DrawString("q=" & kontinuirano1.intenzitet & "[kN/m]", myFont, BrushBrojke, pomocna2 + 5, 15)
End If

```

```

'CRTANJE REAKCIJSKIH SILA
PenSila = New Pen(Brushes.Red, 2.5)
BrushBrojke = New Drawing.SolidBrush(Color.Red)
If ukljestenje Then

    If momentA <> 0 Then
        myGraphics.DrawImage(ImageList2.Images(4), -5, 40)
        myGraphics.DrawString("M", myFont, BrushBrojke, 0, 58)
    End If
    If reakcijaAy <> 0 Then
        pomocna = 20
        myGraphics.DrawLine(PenSila, pomocna, 85, pomocna, 40)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna + 5, 60)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna - 5, 60)
        myGraphics.DrawString("FAy", myFont, BrushBrojke, pomocna + 5, 75)
        myGraphics.DrawLine(PenPomocna, pomocna, 0, pomocna, 900)
    End If
    If reakcijaAx <> 0 Then
        pomocna = 20
        myGraphics.DrawLine(PenSila, 0, 40, pomocna, 40)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna - 5, 45)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna - 5, 35)
        myGraphics.DrawString("FAx", myFont, BrushBrojke, 0, 23)
    End If
Else
    pomocna = (pozicijaA * mjeriloX) + 20

    If pomocni_oslonac_lijevo Then
        myGraphics.DrawLine(PenSila, pomocna, 85, pomocna, 40)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna + 5, 50)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna - 5, 50)
        myGraphics.DrawString("FAy", myFont, BrushBrojke, pomocna + 5, 75)
        myGraphics.DrawLine(PenPomocna, pomocna, 0, pomocna, 900)
    Else
        myGraphics.DrawLine(PenSila, pomocna, 85, pomocna, 40)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna + 5, 50)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna - 5, 50)
        myGraphics.DrawString("FAy", myFont, BrushBrojke, pomocna + 5, 75)
        myGraphics.DrawLine(PenSila, 0, 40, pomocna, 40)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna - 5, 45)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna - 5, 35)
        myGraphics.DrawString("FAx", myFont, BrushBrojke, 0, 23)
        myGraphics.DrawLine(PenPomocna, pomocna, 0, pomocna, 900)
    End If

    pomocna = (pozicijaB * mjeriloX) + 20
    If pomocni_oslonac_desno Then
        myGraphics.DrawLine(PenSila, pomocna, 85, pomocna, 40)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna + 5, 50)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna - 5, 50)
        myGraphics.DrawString("FBy", myFont, BrushBrojke, pomocna + 5, 75)
        myGraphics.DrawLine(PenPomocna, pomocna, 0, pomocna, 900)
    Else
        myGraphics.DrawLine(PenSila, pomocna, 85, pomocna, 40)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna + 5, 50)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna - 5, 50)
        myGraphics.DrawString("FBy", myFont, BrushBrojke, pomocna + 5, 75)
        myGraphics.DrawLine(PenSila, pomocna + 20, 40, pomocna, 40)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna + 5, 45)
        myGraphics.DrawLine(PenSila, pomocna, 40, pomocna + 5, 35)
    End If

```

```

myGraphics.DrawString("FBx", myFont, BrushBrojke, pomocna, 23)
myGraphics.DrawLine(PenPomocna, pomocna, 0, pomocna, 900)
End If
End If
Dim stopX As Single 'zadnja tocka x
Dim stopY As Single 'zadnja tocka y
Dim a As Single 'duljina nosaca
Dim s1 As Single 'pozicija sile 1
Dim s2 As Single 'pozicija sile2
Dim s3 As Single 'pozicija sile3
Dim s4 As Single 'pozicija sile4
Dim rA As Single 'pozicija reakcije a
Dim rB As Single 'pozicija reakcije b
Dim k1 As Single 'pozicija početka kontinuiranog1
Dim k11 As Single 'pozicija kraja kontinuiranog
Dim m1 As Single 'pozicija momenta1
Dim m2 As Single 'pozicija momenta2
Dim pomocna1 As Single
a = duljina_nosaca * mjeriloX
s1 = sila1.pozicija * mjeriloX
s2 = sila2.pozicija * mjeriloX
s3 = sila3.pozicija * mjeriloX
s4 = sila4.pozicija * mjeriloX
k1 = kontinuirano1.pozicija_od * mjeriloX
k11 = kontinuirano1.pozicija_do * mjeriloX
m1 = moment1.pozicija * mjeriloX
m2 = moment2.pozicija * mjeriloX
rA = pozicijaA * mjeriloX
rB = pozicijaB * mjeriloX
'crtanje x osi grafa
Dim PenGraf As Pen
PenGraf = New Pen(Brushes.Black, 9)
pomocna = duljina_nosaca * mjeriloX + 20
myGraphics.DrawLine(PenGraf, 20, 330, pomocna, 330) 'crtanje x osi grafa
pomocniString = "[M]"
myGraphics.DrawString(pomocniString, myFont, BrushBrojke, 0, 320)
stopX = 20
stopY = 330
PenGraf = New Pen(Brushes.Blue, 3)
For i = 1 To 11 Step 1
    If moments(i).pozicija = sila1.pozicija Or moments(i).pozicija = sila2.pozicija Or moments(i).pozicija = sila3.pozicija Or
moments(i).pozicija = sila4.pozicija Or moments(i).pozicija = pozicijaA Or moments(i).pozicija = pozicijaB Or moments(i).pozicija =
moment1.pozicija Or moments(i).pozicija = moment2.pozicija Or moments(i).pozicija = kontinuirano1.pozicija_od Or moments(i).pozicija =
kontinuirano1.pozicija_do Then
        pomocna = moments(i).pozicija * mjeriloX + 20
        pomocna1 = -moments(i).intenzitet * mjeriloY + 330
        Dim trenutniMax As Single
        Dim pozTrenutnogMax As Single
        Dim stvarniMax As Single
        Dim pozStvarnogMax As Single
        Dim tenzijaKrivulje
        stvarniMax = iznosMax
        pozStvarnogMax = mjestoMax
        trenutniMax = 0
        pozTrenutnogMax = 0
        Dim max As Single
        Dim maxpoz As Single
        If kontinuirano1.pozicija_od <= moments(i - 1).pozicija And kontinuirano1.pozicija_do >= moments(i).pozicija And
kontinuirano1.intenzitet <> 0 Then

```

```

20) Then
    If pozStvarnogMax > (moments(i).pozicija * mjeriloX + 20) Or pozStvarnogMax < (moments(i - 1).pozicija * mjeriloX +
    'u slučaju da je mJMax izvan djela za koji crtamo
    pozTrenutnogMax = ((moments(i).pozicija + moments(i - 1).pozicija) / 2) * mjeriloX + 20      ''računaj mJMax
    Dim dio = moments(i).pozicija - moments(i - 1).pozicija      ''dio za koji
racunamo max

    Dim temp As Double = (pozTrenutnogMax - 20) / mjeriloX
    computeMoment(temp, trenutniMax)
    If moments(i - 1).pozicija = moments(i).pozicija Then
        max = pomocna1
    Else max = 330 - trenutniMax * mjeriloY
    End If
    maxpoz = pozTrenutnogMax
Else
    max = 330 - stvarniMax * mjeriloY
    maxpoz = pozStvarnogMax
End If
If maxpoz >= (((moments(i).pozicija + moments(i - 1).pozicija) * mjeriloX) / 2) Then
    tenzijaKrivulje = ((moments(i).pozicija * mjeriloX + 20) - maxpoz) / ((moments(i).pozicija - moments(i -
1).pozicija) * mjeriloX)
ElseIf maxpoz < (((moments(i).pozicija + moments(i - 1).pozicija) * mjeriloX) / 2) Then
    tenzijaKrivulje = (maxpoz - (moments(i - 1).pozicija * mjeriloX + 20)) / ((moments(i).pozicija - moments(i -
1).pozicija) * mjeriloX)
End If
Dim points() As Point = {New Point(stopX, stopY), New Point(maxpoz, max), New Point(pomocna, pomocna1)}
myGraphics.DrawCurve(PenGraf, points, tenzijaKrivulje)
myGraphics.DrawString(((330 - max) / mjeriloY).ToString("0.000"), myFont, BrushBrojke, maxpoz, max - 10)
Else
    myGraphics.DrawLine(PenGraf, stopX, stopY, pomocna, pomocna1)
End If
pomocniString = moments(i).intenzitet.ToString("0.000")
myGraphics.DrawString(pomocniString, myFont, BrushBrojke, pomocna, pomocna1 - 20)
stopX = pomocna
stopY = pomocna1
If moments(i).pozicija = moment1.pozicija Then
    If moment1.pozicija = sila1.pozicija Xor moment1.pozicija = sila2.pozicija Xor moment1.pozicija = sila3.pozicija Xor
moment1.pozicija = sila4.pozicija Xor moment1.pozicija = kontinuirano1.pozicija_do Xor moment1.pozicija = kontinuirano1.pozicija_od Xor
moment1.pozicija = pozicijaA Xor moment1.pozicija = pozicijaB Then
    Else
        If moment1.pozicija = duljina_nosaca Then
        Else
            pomocna1 = stopY + moment1.intenzitet * mjeriloY
            myGraphics.DrawLine(PenGraf, stopX, stopY, stopX, pomocna1)
            stopY = pomocna1
        End If
    End If
End If
If moments(i).pozicija = moment2.pozicija Then
    If moment2.pozicija = sila1.pozicija Xor moment2.pozicija = sila2.pozicija Xor moment2.pozicija = sila3.pozicija Xor
moment2.pozicija = sila4.pozicija Xor moment2.pozicija = kontinuirano1.pozicija_do Xor moment2.pozicija = kontinuirano1.pozicija_od Xor
moment2.pozicija = pozicijaA Xor moment2.pozicija = pozicijaB Then
    Else
        If moment2.pozicija = duljina_nosaca Then
        Else
            pomocna1 = stopY + moment2.intenzitet * mjeriloY
            myGraphics.DrawLine(PenGraf, stopX, stopY, stopX, pomocna1)
            stopY = pomocna1
        End If
    End If
End If
End If
End If

```

```

Next
End Sub
Sub BubbleSort(moments)
Dim i, j As Integer
Dim n = 11
For i = n - 1 To 0 Step -1
    For j = 0 To i - 1
        If moments(j).pozicija > moments(j + 1).pozicija Then
            Call zamjena(moments(j), moments(j + 1))
        End If
    Next j
Next i
End Sub
Sub zamjena(ByRef a As momenti, ByRef b As momenti)
Dim temp As momenti
temp = a
a = b
b = temp
End Sub
Sub computeMoment(temp, ByRef trenutniMax)
pomocna = 0
If sila1.intenzitet <> 0 And sila1.pozicija <= temp Then
    pomocna = pomocna + sila1.intenzitet * Math.Sin(sila1.kut) * (temp - sila1.pozicija)
End If
If sila2.intenzitet <> 0 And sila2.pozicija <= temp Then
    pomocna = pomocna + sila2.intenzitet * Math.Sin(sila2.kut) * (temp - sila2.pozicija)
End If
If sila3.intenzitet <> 0 And sila3.pozicija <= temp Then
    pomocna = pomocna + sila3.intenzitet * Math.Sin(sila3.kut) * (temp - sila3.pozicija)
End If
If sila4.intenzitet <> 0 And sila4.pozicija <= temp Then
    pomocna = pomocna + sila4.intenzitet * Math.Sin(sila4.kut) * (temp - sila4.pozicija)
End If
If kontinuirano1.intenzitet <> 0 And kontinuirano1.pozicija_od < temp Then
    If kontinuirano1.pozicija_do < temp Then
        pomocna = pomocna + kontinuirano1.intenzitet * (vel_kont) * ((vel_kont / 2) + (temp - kontinuirano1.pozicija_do))
    ElseIf kontinuirano1.pozicija_do > temp Then
        pomocna = pomocna + kontinuirano1.intenzitet * (temp - kontinuirano1.pozicija_od) * ((temp - kontinuirano1.pozicija_od) /
2)
    End If
End If
If moment1.intenzitet <> 0 And moment1.pozicija <= temp Then
    pomocna = pomocna - moment1.intenzitet
End If
If moment2.intenzitet <> 0 And moment2.pozicija <= temp Then
    pomocna = pomocna - moment2.intenzitet
End If
If ukljestenje = True Then
    pomocna = pomocna - momentA
End If
If pozicijaA <= temp Then
    pomocna = pomocna + reakcijaAy * (temp - pozicijaA)
End If
If pozicijaB <= temp Then
    pomocna = pomocna + reakcijaBy * (temp - pozicijaB)
End If
trenutniMax = pomocna
End Sub
End Class

```

ZAKLJUČAK

Razvoj ovog programa bio je za mene potpuno novo iskustvo, bilo je to prvo samostalno sastajanje s programskim jezikom Visual Basic. Tijekom izrade nailazio sam na dosta poteškoća koje su zahtjevale online istraživanja i proučavanja. Izradom ovog programa dosta sam napredovao i naučio ponešto o programiranju za platformu Windows.

Uspoređujući rješenja zadataka koji su riješeni ručno uz pomoć kalkulatora i programska rješenja ovog programa dolazimo do zaključka da izrađeni program može služiti kao kalkulator za jednostavne gredne nosače.

Trudit ću se dalje učiti programiranju i razvijati slične aplikacije za pomoć pri učenju za neke druge platforme poput Androida jer smatram da će iste na taj način biti puno pristupačnije korisnicima zbog rapidnog porasta u korištenju smartphone uređaja.

Literatura

- [1] Damjanović, S., Katanić, P., (2014), *Programski jezik Visual Basic Zbirka*. Fakultet poslovne ekonomije, Bijeljina.
- [2] Korporacija Microsoft, Preveo s engleskog Daubachy, D. (1999). *Microsoft Visual Basic 6.0, Vodič za programere*. Zagreb: Strijelac
- [3] Mufić, O. (1991). *Mehanika I (Statika)*. (Teće izdanje), Zagreb: Tehnička Knjiga.

Elektronički izvori:

- [4] Voon Kiong, L. (2006). *Visual Basic 2015 Tutorial*. Preuzeto, 01.07.2017. s <http://www.vbtutor.net/index.php/visual-basic-2015-tutorial/>
- [5] Korporacija Microsoft. *Visual Basic Guide*. Preuzeto, 01.08.2017. s <https://docs.microsoft.com/en-us/dotnet/visual-basic/>
- [6] Allain, A. *Sorting Algorithms Compared* Preuzeto, 05.08.2017. s <http://www.cprogramming.com/tutorial/computersciencetheory/sortcomp.html>
- [7] Matejiček, F. Vnućec, Z. "On Line" *Udžbenik (statika)*. Preuzeto, 01.08.2017. s http://www.sfsb.unios.hr/ksk/statika/prim_mehanika/index.php?izbor=sta&jezik=hr
- [8] Muić, G. i Jurak, M. Prirodoslovno matematički fakultet Zagreb, *Osnove Objektno orjentiranog programiranja*, Preuzeto 05.08.2017. s <https://web.math.pmf.unizg.hr/nastava/rp2/pred2/pred2.html>
- [9] Šutalo, S. Grundler, D. *Osnove programiranja u jeziku C++* Preuzeto 05.08.2017. s <https://sites.google.com/site/sandasutalo/osnove-programiranja/ulazni-i-izlazni-tokovi>