

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

THE SOLVING SIMPLE STATIC BEAM PROBLEM IN THE VISUAL BASIC PROGRAMMING LANGUAGE

Maliković Marko¹, Fumić Robert², Purković Damir²

¹Filozofski fakultet, Sveučilište u Rijeci, Hrvatska

²Studij politehnike, Sveučilište u Rijeci, Hrvatska

SAŽETAK

Učenje temeljnih inženjerskih znanja i vještina danas često pretpostavlja i aktivnosti studenata na povezivanju spoznaja iz različitih polja tehnike, ali i iz drugih područja. Tako se i učenje statike može povezati s različitim računalnim aplikacijama i simulacijama. U ovom radu stoga elaboriramo primjenu programskog jezika Visual Basic za rješavanje problema iz statike. Pri tom predstavljamo razvijenu aplikaciju i primjer rješavanja statičkog problema jednostavnog grednog nosača. Za razliku od besplatnih inačica studentima dostupnih alata za statičke proračune, razvijena aplikacija nije proračunski ograničena. Na primjeru ujedno pokazujemo integriranje sadržaja iz različitih predmeta pri čemu izrada vlastite simulacije u nekom programskom jeziku može značajno unaprijediti učenje statike i programiranja.

Ključne riječi: *gredni nosač, inženjersko obrazovanje, statika, programiranje, Visual Basic*

ABSTRACT

Learning basic engineering knowledge and skills today often presupposes student activities to connect knowledge from different fields of technology, but also from other areas. Therefore, learning of statics also often linked to various computer applications and simulations. This paper elaborates on the application of the Visual Basic programming language for solving simple static problem. A developed application for an example of solving a static problem of a simple pinned beam also presented here.

Unlike the free versions of static calculation tools available to students, the developed application is not feature limited. The example also shows how to integrate content from different subjects, where the creation of their own simulation in a programming language can significantly improve the learning of statics and programming.

Keywords: *pinned beam, engineering education, statics, programming, Visual Basic*

1. UVOD

1. INTRODUCTION

Pri učenju tehnike i inženjerstva danas je važno povezivati znanja iz različitih područja, jer postoji određena praznina između kompetencija koje studenti steknu tijekom studija i onih koje trebaju tržištu rada [1]. Zbog toga obrazovanje inženjera traži brojne kompetencije koje nisu samo povezane s uskim znanjima iz nekog predmeta. U ove kompetencije se ubrajaju: komunikacija i timski rad; profesionalizam i stavovi poput integriteta i predanosti; sposobnosti učenja; upravljanje, usredotočenost na klijenta i poslovne vještine; rukovođenje; prikupljanje i analiza informacija; i interdisciplinarni pristup [2]. Interdisciplinarni pristup, koji ujedno podržava timski rad i međusobnu komunikaciju među studenima, u ovom radu smo implementirali za rješavanje problema kojim se povezuju znanja iz područja statike i programiranja. Na taj način cilj rada je pokazati primjer mogućeg integriranja znanja, kao putokaz studentima i nastavnicima, kao poticaj za implementaciju takvog pristupa u nastavi.

Koristeći besplatne on-line alate poput *SkyCiv Cloud Engineering Software* [3], koji u svojoj besplatnoj verziji služe kao kalkulatori za gredne nosače, naišli smo na problem nemogućnosti unosa opterećenja pod kutom različitim od 90° , kao i nemogućnost unosa više od dvije koncentrirane sile. Također, motivirani problemima koje studenti imaju sa statikom i rješavanjem zadataka koji su temelj tehničke struke odlučili smo izraditi program koji će za uneseni statički određen nosač vratiti numerička rješenja i grafički prikazati opterećenja nosača.

Program smo kreirali u programskom jeziku Visual Basic u kojem smo bili u mogućnosti izračunati potrebne jednadžbe uvjeta ravnoteže čime se dolazi do iznosa reakcijskih sila za odabrane oslonce i iznose momenta fleksije za svaku točku u kojoj se nalazi opterećenje. Također, programski jezik Visual Basic nam je omogućio programiranje grafičkog prikaza pa naš program prikazuje grafikone dobivene iz jednadžbi ravnoteže za opterećenja koja imaju komponente na x osi nosača i za opterećenja koja imaju komponente na y osi nosača. Koristeći iznose momenata fleksije program kreira i grafikon momenata fleksije.

2. PROBLEM GREDNI NOSAČ

2. PINNED BEAM PROBLEM

Gredni nosači ili ravni puni nosači su ustvari uzdužne linije koje povezuju sva težišta poprečnih presjeka nosača, leže u jednoj ravnini, a okomiti su na sve poprečne presjeke te nemaju diskontinuiteta (prekida) između dva susjedna poprečna presjeka nosača [4]. Ovakvi nosači često su sastavni dio mehaničkih konstrukcija. Gredni nosači su sklopovi koji kada ih opteretimo prenose ta opterećenja na oslonce. Oslonci mogu biti pomični, nepomični i uklještenje. Ovisno o vrsti oslonca, pod opterećenjem nosača isti poprimaju vrijednosti reakcijskih sila. 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.

Kod nepomičnog oslonca javljaju se reakcijske sile na apscisi i na ordinati. Prilikom uklještenja oslonac poprima reakcije na apscisi, ordinati i dodatno se javlja reakcijski moment koji prikazuje djelovanje sile na kraku. Nosače možemo opteretiti silama, kontinuiranim opterećenjima i 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 njutnima [N] ili kilonjutnima [kN]. Kontinuirana opterećenja karakteriziraju dvije različite točke, na udaljenosti od lijevog kraja nosača, također izraženom u metrima. Između tih točaka, nalazi se kontinuirano opterećenje koje ima jednu komponentu po y osi izraženu u njutnima po metru [N/m] ili kilonjutnima po metru [kN/m]. Momenti su također karakterizirani udaljenošću od lijevog kraja nosača, oni nastoje zarotirati nosač, a intenzitet im je izražen u njutnmetrima [Nm] ili kilonjutnmetrima [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. Drugim riječima, reakcije u osloncima moraju biti takve da sa zadanim opterećenjem nosača čine ravnotežu sila [5].

3. PROGRAMSKI JEZIK VISUAL BASIC

3. VISUAL BASIC PROGRAMMING LANGUAGE

Visual Basic je objektno orijentiran programski jezik za izradu programa pogonjenih događajima (neka operacija se vrši ako korisnik želi da bude izvršena odnosno ako korisnik nekom akcijom potakne izvršavanje te operacije, npr. klikom na gumb). Programski jezik Visual Basic je razvila tvrtka Microsoft, te zbog toga velik broj korisnika platforme Windows koristi Visual Basic za razvoj raznih programa i aplikacija. Visual Basic nastao je na temeljima programskog jezika Basic [6].

Slika 1 Kartica za postavljanje nosača, Izvor: autori

Figure 1 Beam setting tab, Source: authors

Visual Basic spada u RAD (Rapid Application Development) programske jezike što znači da je zamišljen kao alat za brzu izradu programa. Ta brzina ostvaruje se tako da se u program dodaju gotovi elementi koji su ugrađeni u sustav Microsoft Windows. Osim što omogućava pisanje programskog kôda, Visual Basic omogućava programerima lako kreiranje GUI (Graphical User Interface) sučelja koje korisniku omogućava jednostavnu interakciju s programom (na primjer klikom na gumb i sličnim jednostavnim akcijama).

4. ULAZNI PODACI

4. INPUT DATA

Ulazni podaci su jedan od ključnih dijelova ovog programa. Pomoću njih korisnik definira veličinu nosača, vrste oslonaca, vrste opterećenja i njihove pozicije. Ti podaci se unose na prvoj kartici programa prikazanoj na Slici 1 i nazvanoj *Kartica za postavljanje nosača*.

Kao što vidimo, za odabir oslonaca koristimo radio gumbe, a za odabir uklještenja okvir za izbor. Nakon toga, klikom na gumb *Generiraj nosač* pokrećemo algoritam za kreiranje nosača i izračun reakcijskih sila kao

i momenta fleksije te pokretanje algoritma za crtanje grafičkih rezultata. Također smo kreirali gumb za ponovno postavljanje (*Reset*) unosa ako želimo s unosom krenuti od početka. U Tablici 1 prikazani su ulazni podaci, nazivi varijabli i tipovi podataka koje smo koristili u programu.

Na drugoj kartici programa prikazanoj na Slici 2 se korisniku omogućava da izabere vrstu i iznos opterećenja kojom želi opteretiti nosač i smjer momenta u smjeru kazaljke na satu ili obrnuto od smjera kazaljke na satu. Korisnik može odrediti kut sile čija početna vrijednost je 90° , a on je može promijeniti u intervalu od 0° do 180° .

Kako se je u ovom dijelu programa ukazala potreba za složenijim tipovima podataka, kreirali smo takozvane strukture. Strukture su složeni tipovi podataka koji mogu sadržavati više različitih tipova podataka [7]. U našem programu smo definirali tri strukture koje se odnose na silu koja se sastoji od intenziteta, pozicije i kuta, kontinuirano opterećenje koje se sastoji od intenziteta, pozicije početka i svršetka opterećenja i moment koji se sastoji od intenziteta, pozicije i smjera. U Tablici 2 prikazani su nazivi struktura, nazivi varijabli u pojedinim strukturama i korišteni tipovi podataka.

Ulazni podatak	Naziv varijable	Tip podatka
Duljina nosača	duljina_nosaca	double
Duljina prepusta	duljina_prepusta	double
Lijevi pomični oslonac	pomicni_oslonac_lijevo	boolean
Desni pomični oslonac	pomicni_oslonac_desno	boolean
Lijevi nepomični oslonac	nepomicni_oslonac_lijevo	boolean
Desni nepomični oslonac	nepomicni_oslonac_desno	boolean
Uklještenje	ukljestenje	boolean
Prepust lijevo	prepust_lijevo	boolean
Prepust desno	prepust_desno	boolean
Prepust obje strane	prepust_obje_strane	boolean

Tablica 1. Ulazni podaci

Table 1. Input data

Izvršiti unos opterećenja
Polja u kojima iznos opterećenja iznosi nula [0] neće se uzeti u obzir.

Iznos opterećenja: Unesi iznos sile u kiloNewton-ima. Pozicija: U metrima. Kut opterećenja: Iznos kuta s obzirom na horizontalnu os x.

Iznos kontinuiranog opterećenja: U kiloNewton-ima po metru. Pozicija od: U metrima. do: U metrima.

Iznos opterećenja momentom: U kiloNewton metrima. Pozicija: U metrima. Smjer: Odaberi...

Izračunaj!

Slika 2 Kartica za unos opterećenja, Izvor: autori

Figure 2 Load input tab, Source: authors

Tablica 2. Strukture

Table 2. Structures

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

5. IZLAZ

5. OUTPUT

Izlazni podaci su ono što čini program korisnim. U izlaznim podacima su sadržana rješenja problema koje nam daje naš program na temelju ispravno unesenih ulaznih podataka i odgovarajućih algoritama. Program nam, osim numeričkih rezultata, koristeći algoritme za crtanje koje smo napisali, kreira i vizualni izlaz. I numerički i grafički izlazni podaci prikazuju se na trećoj kartici koju vidimo na Slici 3.

Prostor za grafički prikaz se proteže preko 80% kartice (prikazani su pomoću algoritma za crtanje na objektu *PictureBox*) dok za numeričke rezultate služi ostatak kartice desno od grafičkog prikaza.



Slika 3 Kartica za prikaz rezultata, Izvor: autori

Figure 3 Results tab, Source: authors

Za momente fleksije korištena je struktura *momenti* koja ima mogućnost pohranjivanja intenziteta momenta fleksije te njegove udaljenosti od lijevog kraja nosača. Nazivi članova strukture i korišteni tipovi podataka prikazani su u Tablici 3.

Tablica 3. Struktura za pohranu momenata fleksije

Table 3. Flexion moments storage structure

Naziv strukture	Naziv člana strukture	Tip podatka
momenti	intenzitet	double
	pozicija	double

6. METODE

6. METHODS

U objektno orijentiranim programskim jezicima u koje spada i Visual Basic, glavnu ulogu imaju objekti koji sadrže podatke i metode. Podaci objekta predstavljaju njegovo stanje dok se ti podaci pomoću metoda mogu mijenjati i komunicirati s drugim objektima [8]. Četiri ključne metode koje koristimo u našem programu se pokreću klikom na gumb *Generiraj nosač* na kartici za postavljanje nosača, na gumb *Izračunaj* na kartici za unos opterećenja te na gumb *Graf. prikaz opterećenja na x i y osi* i *Graf. prikaz momenata fleksije* na kartici za prikaz rezultata.

Kada se pokrene metoda iz gumba *Generiraj nosač*, ona najprije traži greške u unosu i to neispravan unos duljine nosača, neispravan unos oslonaca, unesen prepust veći od ukupne duljine nosača i ako korisnik odabere statički neodređen slučaj koristeći dva nepomična oslonca. U tim slučajevima program javlja korisniku da nije moguće izvesti generiranje nosača. U slučaju kada je unos ispravan, ova metoda provjerava postoje li dva pomična oslonca kako bi se mogao prikazati shematski prikaz kreiranog nosača. U slučaju nosača s pomičnim osloncima ne smijemo opteretiti isti s opterećenjima koja imaju komponentu na apscisi. Zato program isključuje mogućnost promjene kuta na kartici za unos opterećenja i postavlja fiksnu vrijednost od 90° , odnosno okomito na nosač.

Zatim korisnik na slijedećoj kartici vrši unos opterećenja koja djeluju na prethodno kreiran nosač, a klikom na gumb *Izračunaj* se pokreće metoda koja ima tri ključna trenutka i koja je od velike važnosti u programu jer se u njoj izračunavaju reakcijske sile, kao i momenti fleksije. U toj metodi se provjerava postoji li kontinuirano opterećenje i ako postoji tada se izračunava njegova duljina. Nadalje, podešavaju se vrijednosti momenata s obzirom na to da li je smjer u smjeru kazaljke na satu ili obrnuto te mu se ovisno o slučaju dodjeljuje predznak.

U nastavku ove metode izračunavaju se reakcije oslonaca, redom za slučajeve s lijevim nepomičnim i desnim pomičnim osloncem, lijevim pomičnim i desnim nepomičnim te oba pomična i uklještenje. Algoritam radi tako da prvo pronađe 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 dobije se i posljednja reakcija. U slučaju uklještenja se javlja još i reakcijski moment koji algoritam računa tako da zbroji momente svih opterećenja za točku u kojoj je uklještenje. Nakon računanja reakcija, vrši se ispis rezultata reakcijskih sila na trećoj kartici programa. Nakon što završi sa izračunavanjem reakcijskih sila, metoda izračunava momente fleksije za svaku točku u kojoj se nalazi neko opterećenje. Ako uzmemo kao primjer izračunavanje momenata fleksije za točku u kojoj se nalazi oslonac A, odnosno lijevi oslonac nosača, tada algoritam provjerava radi li se o uklještenju i ako je tako, moment fleksije za tu točku poprima vrijednost reakcijskog momenta, a zatim algoritam provjerava postoji li prepust lijevo od točke za koju računamo moment fleksije. U slučaju kada prepust ne postoji, moment fleksije automatski poprima vrijednost nula jer znamo da je vrijednost momenta fleksije u krajevima nosača jednak nuli. U suprotnom, algoritam ispituje postoji li opterećenje koje je lijevo od našeg oslonca. Ako postoji, izračunava se moment fleksije za tu komponentu i pribraja se ostatku. Analogno rade programski kôdovi za sva opterećenja koja se mogu naći na nosaču.

Nakon izračunavanja reakcijskih sila, momenata i momenata fleksije na red dolazi metoda za crtanje grafičkog prikaza koja se pokreće klikom na gumb *Graf. prikaz opterećenja na x i y osi* na kartici za prikaz rezultata.

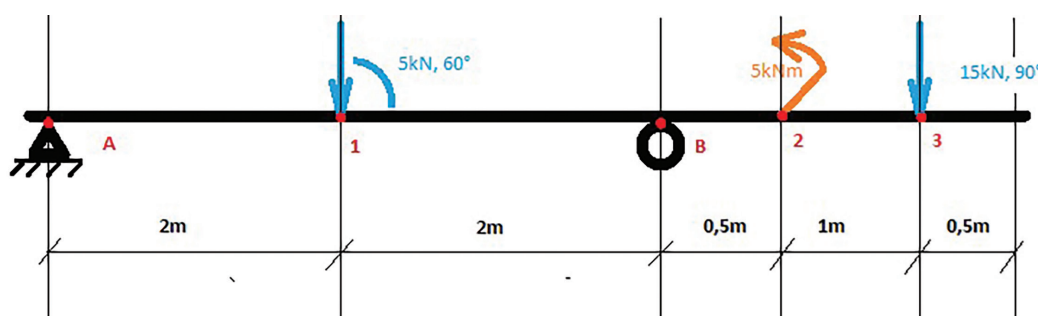
Ova metoda provjerava postoji li horizontalno opterećenje. Ako postoji, pokreće se algoritam za crtanje grafova za horizontalna opterećenja.

Ispod gumba za crtanje grafičkih prikaza se nalazi gumb za crtanje momenata fleksije. Prethodno smo, prilikom računanja numeričkih rezultata, izračunali moment fleksije za svaku točku u kojoj postoji opterećenje. Ti momenti se u ovom trenutku programa nalaze u jednodimenzionalnom polju 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. Zato program najprije sortira članove tog polja. Rekli smo da struktura *momenti* ima članove *pozicija* i *intenzitet* koji zapravo predstavljaju koordinate grafa momenata fleksije. Član *pozicija* predstavlja x koordinatu dok *intenzitet* predstavlja y koordinatu te tih dvanaest vrijednosti tvori poligon točaka grafa. U crtežu je potrebno ravnim linijama spojiti te ključne točke sortiranog niza, odnosno zakrivljenim linijama ako je u pitanju kontinuirano opterećenje.

7. PRIMJER KONKRETNOG PROBLEMA

7. AN EXAMPLE OF A SPECIFIC PROBLEM

U ovom poglavlju prikazujemo rješenje jednog jednostavnijeg problema najprije kroz proračun "na papiru", a zatim to rješenje uspoređujemo s programskim rješenjima za isti slučaj. Kao što vidimo na Slici 4, u točki A koja se nalazi na početku nosača, postavljen je nepomični oslonac što znači da je statički određen reakcijskim silama po x i y osi.



Slika 4 Skica konkretnog problema, Izvor: autori

Figure 4 Specific problem sketch, Source: authors

U točki B postavljen je pomični oslonac, pa je od statički određen silom reakcije samo po y osi. Izračunajmo ih koristeći uvjete

$$\sum F_x = 0, \sum F_y = 0, \sum M_A = 0.$$

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

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

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

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

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

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

$$Mf_A = 0$$

$$Mf_1 = F_{Ay} \cdot 2m = -4,42 \text{ kNm}$$

$$Mf_B = F_{Ay} \cdot 4m - F_1 \cdot \sin(60^\circ) \cdot 2m = -17,5 \text{ kNm}$$

$$Mf_2 \text{ bez mometa} = F_{Ay} \cdot 4.5m - F_1 \cdot \sin(60^\circ) \cdot 2.5m + F_{By} \cdot 0.5m = -10 \text{ kNm}$$

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

$$Mf_3 = F_{Ay} \cdot 5.5m - F_1 \cdot \sin(60^\circ) \cdot 3.5m + F_{By} \cdot 1.5m - M_1 = 0 \text{ kNm}$$

Da bismo ovaj problem riješili koristeći naš program, najprije unosimo ulazne podatke. Prvo odredimo da je nosač veličine šest metara, pridružimo mu nepomični oslonac u lijevom kraju nosača te pomični u desnom kraju uz prepust od 2 metra (slika 5).

Nakon što kliknemo na gumb *Generiraj nosač*, kontrola programa prijeđe na karticu za unos opterećenja (slika 6).

U polja za prvu silu unosimo redom:

- iznos opterećenja = 5 kN
- pozicija = 2 m
- kut opterećenja = 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 je obrnut od kazaljke na satu

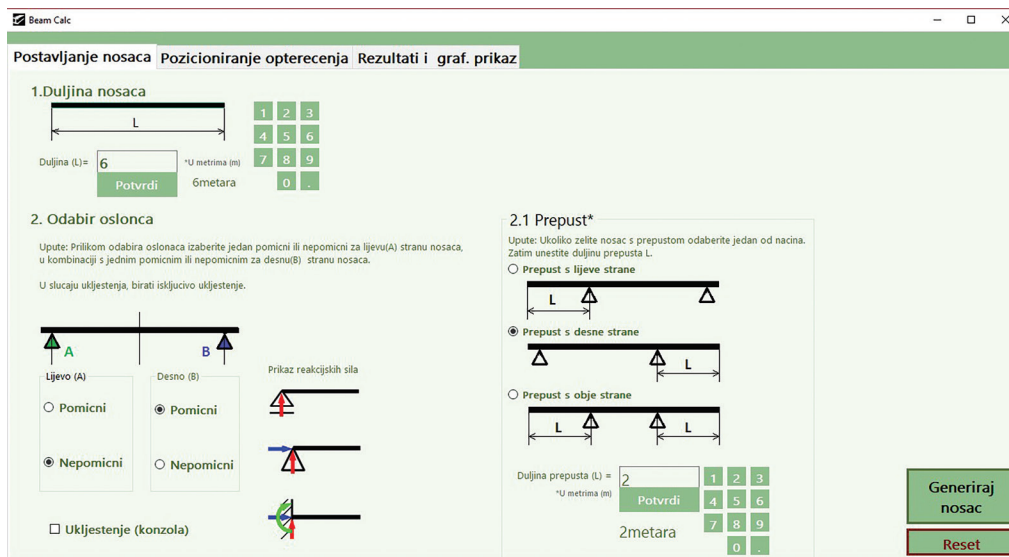
Klikom na gumb *Izračunaj*, program prelazi na karticu s rješenjima (Slika 7).

Iz Slike 7 možemo vidjeti da program daje jednaka rješenja kao što smo ih dobili i "ručno". Na gornjem od dva grafa prikazano je opterećenje na x osi nosača, a kako samo dvije komponente imaju opterećenje na x osi on je vrlo jednostavan. S lijeva na desno je nacrtan iznos reakcije F_{Ax} vertikalno, zatim slijedi horizontalna linija bez promjene sve do opterećenja F_1 gdje je nacrtana njena horizontalna komponenta vertikalno i linija završava u nultoj vrijednosti. Na donjem grafu prikazano je opterećenje y osi nosača te je uočljivo kako se graf mijenja nailaskom na opterećenja.

Redom je nacrtan F_{Ay} vertikalno, zatim slijedi horizontalna linija bez promjene sve do opterećenja F_1 gdje ponovo ostavlja trag u obliku vertikalne linije veličine iznosa opterećenja $F_1 \cdot \sin(60^\circ)$, zatim horizontalnom linijom bez promjene nastavlja sve do F_{By} gdje crta njen iznos vertikalno. Na kraju slijedi horizontalna linija do mjesta opterećenja F_2 čiji iznos se crta vertikalno te graf završava u nultoj vrijednosti.

Na Slici 8 možemo vidjeti još graf momenata fleksije. U jednodimenzionalnom polju strukture momenti je pohranjen poligon točaka ovog grafa kojeg jednostavno spojimo odgovarajućim linijama.

Moment fleksije u početku nosača iznosi nula, što je ujedno i iznos momenta fleksije za oslonac A. U drugoj točki, odnosno, na poziciji sile F_1 moment fleksije iznosi -4.42 kNm. Naš program spaja te dvije točke linijom te nastavlja dalje do oslonca B u čijoj poziciji moment fleksije iznosi -17.5 kNm. Nakon tog dolazimo u točku momenta 1 u kojoj je iznos momenta fleksije -10 kNm pa algoritam za crtanje grafa od te točke nanosi vertikalno iznos momenta u iznosu od 5 kNm, čime dolazi do vrijednosti 15 kNm iz "ručnog" proračuna. Moment fleksije za poziciju sile F_2 iznosi 0 kNm, pa linija grafa dolazi u nultu i krajnju vrijednost.

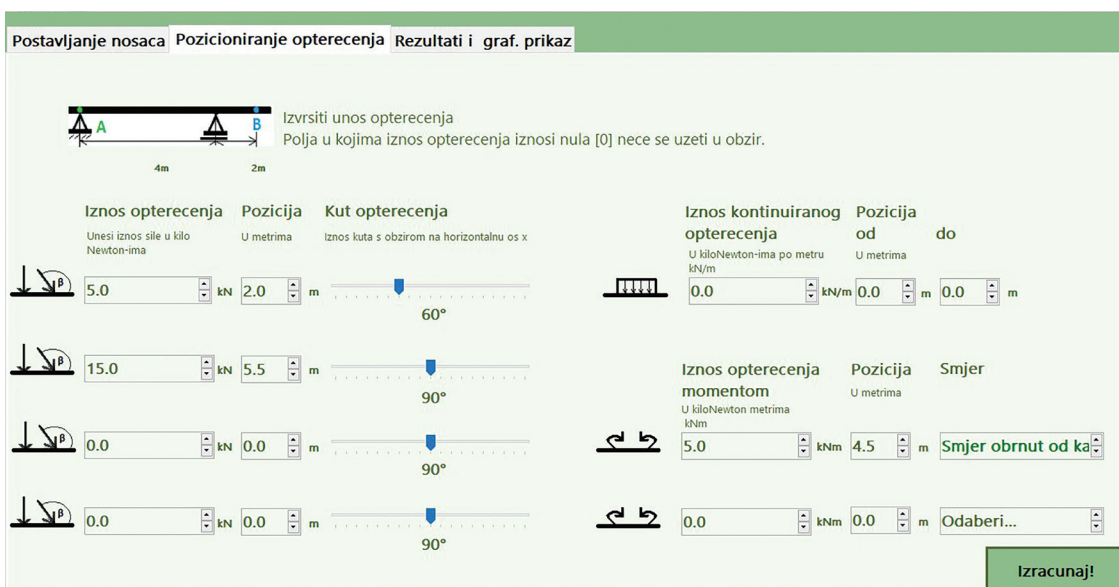


Slika 5 Unos ulaznih podataka za konkretan problem, Izvor: autori

Figure 5 Enter input for a specific problem, Source: authors

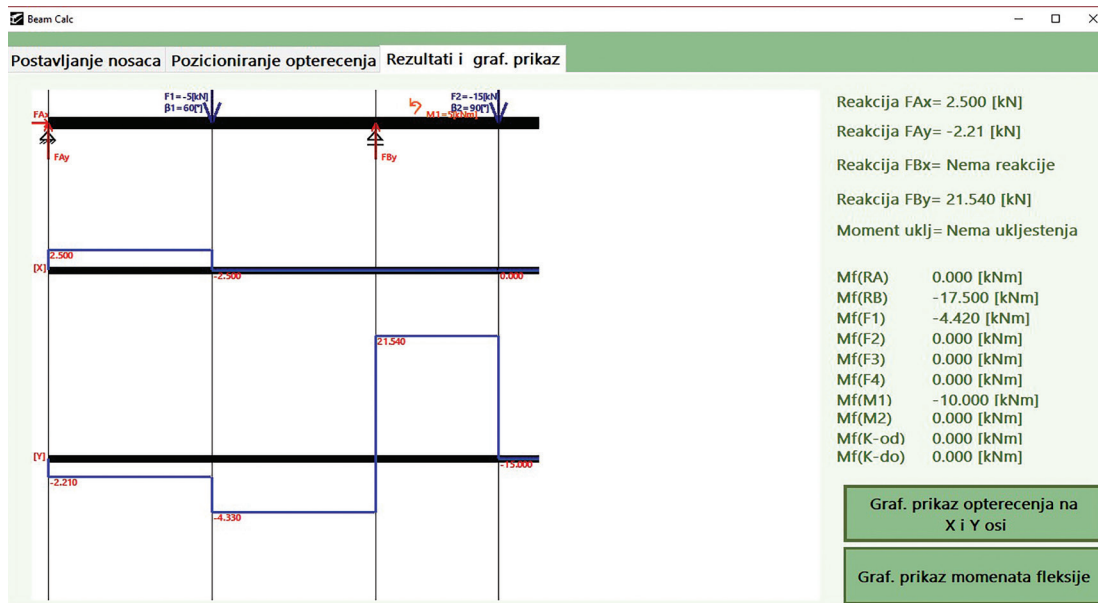
Slika 6 Unos opterećenja za konkretan problem, Izvor: autori

Figure 6 Enter input for a specific problem, Source: authors



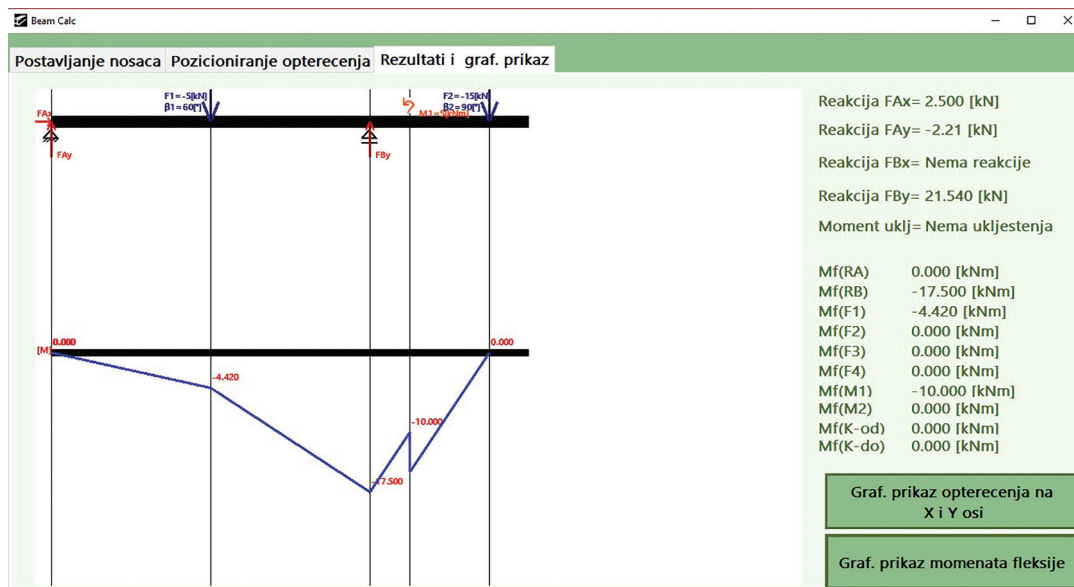
Slika 7 Rezultati i grafički prikaz za konkretan problem, Izvor: autori

Figure 7 Results and graphical presentation for a specific problem, Source: authors



Slika 8 Graf momenata fleksije za konkretan problem, Izvor: autori

Figure 8 Flexion moments graph for a specific problem, Source: authors



8. ZAKLJUČAK

8. CONCLUSION

Razvoj cjelovitog programa za primjenu u tehničkim strukama je za studente takvih struka često potpuno novo ali pozitivno iskustvo. Naime, iako studenti tehničkih struka u pravilu uče programirati, samo rijetki dođu u fazu da stečena znanja zaokruže u cjelinu kao što je potpuno funkcionalan program koji ima svoju primjenu.

Ako se to ipak dogodi, tada izradom nekog takvog programa studenti obavezno veoma dobro napreduju i nauče mnogo o programiranju. Jasno je da obavezno nauče mnogo i o struci na koju se program odnosi.

Osim toga, u konkretnom slučaju programa opisanog u ovom radu, uspoređujući rješenja zadataka dobivenih ručno uz pomoć kalkulatora i programska rješenja našeg programa možemo zaključiti da izrađeni program može služiti kao kalkulator za jednostavne gredne nosače pa samim time i kao pomoć studentima u proračunima.

Zbog svega navedenog, mišljenja smo da je potrebno i dalje kreirati i razvijati nove programe koji će služiti kao pomoć pri učenju i to za različite platforme. Na primjer, programi za platforme poput Androida bi mogle biti puno pristupačnije korisnicima zbog rapidnog porasta u korištenju pametnih telefona.

9. REFERENCE

9. REFERENCES

- [1.] Abdulwahed M.; Hasna M. O.; Engineering and Technology Talent for Innovation and Knowledge-Based Economies: Competencies, Leadership, and a Roadmap for Implementation; Springer International Publishing AG, 2017.
- [2.] Male S. A.; Generic Engineering Competences; A Review and Modelling Approach, Education research and Perspectives, Vol. 37., No. 1, 2010.
- [3.] SkyCiv Engineering; SkyCiv Cloud Engineering Software, <https://skyciv.com>.
- [4.] Matejiček, F.; Vnućec, Z.; Statika – elektronički udžbenik, <https://www.sfsb.hr/ksk/statika>.
- [5.] Muftić O.; Mehanika I (Statika), treće izdanje; Zagreb: Tehnička Knjiga, 1991.
- [6.] Microsoft; Tutorial: Get started with Visual Basic in Visual Studio, <https://docs.microsoft.com/en-us/visualstudio/get-started/visual-basic/>.
- [7.] Microsoft; Visual Basic Guide, <https://docs.microsoft.com/en-us/dotnet/visual-basic/>.
- [8.] McMillan M.; Object-Oriented Programming with Visual Basic.NET; Cambridge University Press, 2004.

AUTORI · AUTHORS

● Maliković Marko

Rođen je 1965. godine. Diplomirao je na Pedagoškom fakultetu u Rijeci, smjer Profesor matematike i informatike, a od 1997. radi na Filozofskom fakultetu u Rijeci. Doktorsku disertaciju obranio je 2008. na Fakultetu organizacije i informatike u Varaždinu iz područja društvenih znanosti, polje informacijskih znanosti, grana informacijski sustavi i informatologija.

Nositelj je više kolegija u okviru različitih studija na Filozofskom fakultetu u Rijeci. Objavio je dvadesetak znanstvenih radova iz područja primjene simboličke logike u računalnim znanostima, automatskog rezoniranja, formalnih sustava, višeagentnih sustava i internetskih istraživanja. Sudjelovao je na dva znanstveno-istraživačka projekta i izlagao desetak radova na međunarodnim znanstvenim skupovima. Autor je sveučilišnog priručnika Internetska istraživanja, kojeg je objavio Filozofski fakultet u Rijeci, 2015. godine. Glavni je urednik znanstvenog časopisa Politehnika: časopis za tehnički odgoj i obrazovanje, izdavač Sveučilište u Rijeci.

Korespondencija · Correspondence

marko@ffri.hr

● Purković Damir

Radio je kao asistent u znanstvenom području društvenih znanosti na Filozofskom fakultetu u Rijeci, Hrvatska, te kao suradnik za metodičku skupinu predmeta na Odsjeku za politehniku. Od 2009. je radio kao predavač iz područja tehničkih znanosti, polje interdisciplinarnih tehničkih znanosti. Doktorirao je na Prirodoslovno-matematičkom fakultetu Sveučilišta u Splitu, u znanstvenom području: interdisciplinarna područja znanosti, polje obrazovne znanosti. Na Sveučilištu u Rijeci izabran je za docenta u interdisciplinarnom području znanosti. Autor je mnogih znanstvenih i stručnih radova iz područja obrazovnih znanosti, inženjerstva i informacijsko-komunikacijskih znanosti. Autor je nekoliko sveučilišnih publikacija i glavni urednik časopisa Politehnika. Kao predavač i pozvani predavač sudjelovao je na mnogim međunarodnim znanstvenim konferencijama..

Korespondencija · Correspondence

damir@uniri.hr

● Fumić Robert

Rođen je 1995. godine u Ogulinu. Preddiplomski sveučilišni studij Politehnike završio je 2017. godine na Studiju politehnike Sveučilišta u Rijeci, a 2020. je diplomirao na sveučilišnom studiju politehnike i informatike u Rijeci.

Korespondencija · Correspondence

robert.fumic123@gmail.com