



.....
Srednja šola za kemijo,
elektrotehniko in računalništvo

Matematični program

Raziskovalna naloga

Avtorja:

Blaž Hribernik in Urban Levičar, R-4.a

Mentor:

mag. Boštjan Resinovič, univ. dipl. inž. rač. in inf.

Mestna občina Celje, Mladi za Celje

Celje, april 2023

"Vsi deli bi se morali brez prisile ujemati skupaj. Spomniti se morate, da ste vi tisti, ki ste jih razstavili. Če jih ne morete ponovno sestaviti, mora obstajati razlog. Nikakor ne uporabljajte kladiva."

— IBM Manual, 1925

ZAHVALA

S to zahvalo bi se rada zahvalila mentorju Boštjanu Resinoviču za njegov dragoceni čas, trud in podporo, ki nama jo je namenil med izvajanjem raziskovalne naloge. Njegovo strokovno znanje in izkušnje so nama bili v veliko pomoč.

Povzetek

Glavni namen raziskovalne naloge je bil razviti program, ki bi pomagal dijakom pri učenju matematike. Pri izdelavi programa sva upoštevala rezultate raziskave, v kateri sva ugotavljala, kaj dijakom pri učenju matematike povzroča največ težav. Program je bil razvit v programskem jeziku C# z uporabo orodja Windows Forms, ki je namenjeno razvoju grafičnih uporabniških vmesnikov.

Med raziskovanjem sva ugotovila, da primanjkuje programov za računanje z liki in telesi v slovenskem jeziku. Čeprav obstajajo nekateri programi, ki omogočajo reševanje matematičnih problemov, so ti večinoma v angleškem jeziku in za manj izkušene uporabnike niso enostavni. Zato sva se odločila, da bova v okviru raziskovalne naloge izdelala program za računanje z liki in telesi ter risanje raznih funkcij, ki bo uporabnikom omogočal enostavno in učinkovito računanje stranic, kotov ter ostalih podatkov za like in telesa glede na uporabnikov vnos. Ustvarila sva tudi spletno stran, na kateri je mogoče reševanje nalog in prenos programa.

Ključne besede: Matematika, C#, Windows Forms, liki, telesa, funkcije, polinomi

Abstract

The main aim of the research paper was to develop a software program that would help students learn mathematics. While designing the program, we considered the results of a survey in which we tried to find out what causes students the most difficulties in learning math. The program was developed in C# using Windows Forms, a tool for developing graphical user interfaces.

During our research, we found out that there is a lack of programs for calculating with figures and solids in the Slovene language. Although there are some programs that allow solving mathematical problems, these are mostly in English and are not easy to use for less experienced users. Therefore, as a part of the research task, we decided to develop a program for calculating with characters and solids and drawing various functions that will allow users to calculate sides, angles, and other data easily and efficiently for characters and solids according to user input. We have also created a web page where the tasks can be solved, and the program can be downloaded.

Keywords: Mathematics, C#, Windows Forms, shapes, bodies, functions, polynomials

Vsebina

| | | |
|----------|---|-----------|
| 1 | Uvod..... | 7 |
| 1.1 | Opredelitev področja in raziskovalnega problema | 7 |
| 1.2 | Cilji raziskovalne naloge | 8 |
| 1.3 | Hipoteze..... | 9 |
| 1.4 | Metodologija..... | 9 |
| 2 | Anketa | 11 |
| 2.1 | Izbor in struktura vzorca anketiranih..... | 11 |
| 2.2 | Rezultati ankete | 12 |
| 3 | Primerjava z že obstoječima programoma | 14 |
| 4 | Program..... | 15 |
| 4.1 | Visual Studio | 15 |
| 4.1.1 | Splošno | 15 |
| 4.1.2 | Windows Forms | 15 |
| 4.2 | C#..... | 16 |
| 4.2.1 | Splošno | 16 |
| 4.2.2 | Zgodovina..... | 16 |
| 4.3 | Knjižnice..... | 17 |
| 4.3.1 | ZedGraph..... | 17 |
| 4.3.2 | Newtonsoft.Json | 17 |
| 4.3.3 | ZXing | 19 |
| 4.3.4 | Otp.NET | 20 |
| 4.4 | Delovanje programa | 21 |
| 4.4.1 | Okno za prijavo | 22 |
| 4.4.2 | Liki | 23 |
| 4.4.3 | Telesa | 28 |
| 4.4.4 | Uporabljene formule | 28 |

| | | |
|----------|----------------------------------|-----------|
| 4.4.5 | Pretvarjanje enot..... | 29 |
| 4.4.6 | Risanje funkcij..... | 29 |
| 4.4.7 | Kalkulator..... | 31 |
| 4.5 | Namestitev | 32 |
| 5 | Spletna stran | 33 |
| 5.1 | Predstavitev | 33 |
| 5.2 | Načrtovanje podatkovne baze..... | 33 |
| 5.3 | Gostovanje | 34 |
| 5.4 | Registracija in prijava..... | 34 |
| 5.5 | Nalaganje programa..... | 35 |
| 5.6 | Reševanje vaj..... | 35 |
| 5.7 | Pregled rešenih vaj..... | 36 |
| 5.8 | Izpis uporabnika | 37 |
| 6 | Ovrednotenje hipotez..... | 38 |
| 7 | Zaključek..... | 40 |
| 8 | Literatura | 41 |
| 9 | Priloge..... | 43 |
| 9.1 | Anketni vprašalnik..... | 43 |

Kazalo slik

| | | |
|----------|--|----|
| Slika 1: | Starost anketirancev. | 11 |
| Slika 2: | Spol anketirancev..... | 12 |
| Slika 3: | Uspeh anketirancev pri matematiki. | 12 |
| Slika 4: | Mnenje o uporabi. | 13 |
| Slika 5: | Uporaba metode JsonConvert.SerializeObject. | 18 |
| Slika 6: | Niz JSON | 18 |
| Slika 7: | Program..... | 21 |
| Slika 8: | Okno za prijavo..... | 22 |

| | |
|--|----|
| Slika 9: Del kode za vpis v program. | 22 |
| Slika 10: Shranjevanje podatkov v lokalno podatkovno bazo. | 23 |
| Slika 11: Podmeni. | 23 |
| Slika 12: Znak za prepovedano. | 24 |
| Slika 13: Okno za poljubni trikotnik. | 24 |
| Slika 14: Izsek kode za preverjanje podatkov pravokotnega trikotnika. | 25 |
| Slika 15: Izsek kazalca in barvo polj. | 25 |
| Slika 16: Pridobivanje podatkov iz polj. | 26 |
| Slika 17: Drevesna struktura razredov. | 26 |
| Slika 18: Razred PravokotniTrikotnik. | 27 |
| Slika 19: Kreiranje pravokotnega trikotnika. | 27 |
| Slika 20: Izračun večkotnika. | 28 |
| Slika 21: Okno za pretvarjanje enot. | 29 |
| Slika 22: Prikaz funkcije. | 30 |
| Slika 23: Okno za risanje grafov. | 30 |
| Slika 24: Funkcije za računanje točk. | 31 |
| Slika 25: Okno kalkulator. | 31 |
| Slika 26: Razširitev "Setup Project" za namestitev programa. | 32 |
| Slika 27: Nameščena datoteka na namizju. | 32 |
| Slika 28: Osnovna stran. | 33 |
| Slika 29: ER diagram podatkovne baze. | 34 |
| Slika 30: Izsek kode za registracijo. | 35 |
| Slika 31: Reševanje vaje. | 36 |
| Slika 32: Uporabnikove rešene vaje. | 36 |
| Slika 33: Analiza rešene vaje. | 37 |
| Slika 34: Izpis uporabnika. | 37 |

1 Uvod

1.1 Opredelitev področja in raziskovalnega problema

V današnjem svetu računalništva se vse bolj poudarja pomen matematičnega znanja in sposobnosti reševanja matematičnih nalog. Zato sva se odločila raziskati področje matematičnih programov, ki bi lahko uporabnikom pomagali pri reševanju kompleksnih matematičnih problemov.

Najina raziskava se osredotoča na razvoj matematičnega programa, ki bi lahko uporabnikom pomagal pri reševanju kompleksnih matematičnih problemov in hkrati izboljšal njihove matematične veščine. Najin cilj je ustvariti program, ki bo uporabnikom omogočil učenje matematike na interaktiven način in uspešno reševanje zahtevnih nalog. S takšnim programom bi lahko pomagala dijakom, profesorjem in vsem tistim, ki želijo izboljšati svoje matematične sposobnosti.

1.2 Cilji raziskovalne naloge

Cilj najine raziskovalne naloge je bil ustvariti matematični program za preverjanje matematičnih nalog in risanje polinomov, ki vključuje tudi formule in razlage zahtevnejših matematičnih snovi. Želela sva ustvariti program, ki bo uporaben tudi za profesorje matematike.

Cilji teoretičnega dela:

- predstaviti osnovne matematične principe, ki so potrebni za razumevanje polinomov in drugih matematičnih konceptov, ki jih bo program vseboval;
- predstaviti matematične formule, ki jih bo program uporabljal za reševanje matematičnih nalog;
- raziskati in predstaviti obstoječe matematične programe za risanje grafov in primerjati njihove funkcionalnosti z najinim programom;
- raziskati in preučiti matematične formule, ki bodo uporabljene za računanje podatkov o likih in telesih;

Cilji praktičnega dela:

- omogočiti računanje podatkov o likih in telesih glede na vpisane podatke uporabnika;
- implementirati program za risanje polinomov in preverjanje matematičnih nalog;
- preizkusiti program s testnimi nizi nalog ter ga optimizirati za najboljše rezultate;
- oblikovati spletno stran, ki bo enostavna za uporabo in prijazna do uporabnikov;
- testirati program med profesorji matematike in pridobiti njihove povratne informacije za morebitne izboljšave;
- dodati funkcionalnost shranjevanja rezultatov preizkusa znanj in izpisovanja grafov, ki jih bo uporabnik narisal;

Skupni cilj te raziskovalne naloge je ustvariti uporabniku prijazen program za risanje polinomov in preverjanje matematičnih nalog ter vzbuditi zanimanje profesorjev matematike za uporabo programa v šolskem okolju.

1.3 Hipoteze

V zvezi z najino raziskavo sva oblikovala in postavila pet hipotez, ki predvidevajo uspešnost in ustreznost najinega matematičnega programa:

H1: Profesorji bodo program prepoznali kot primeren za poučevanje.

H2: Program bo omogočal reševanje matematičnih nalog, kot so geometrija, trigonometrija, polinomi, ...

H3: Za pravilno delovanje programa, bodo morale biti vključene različne knjižnice.

H4: Za izdelavo programa bo zadoščalo znanje, pridobljeno v šoli.

H5: Stroški bodo nastali samo pri gostovanju spletne strani.

1.4 Metodologija

Začela sva s postopkom raziskovanja, tako da sva najprej določila cilje in hipoteze raziskovalne naloge. Cilj je bil razviti matematični program, ki bi pomagal uporabnikom pri preverjanju matematičnih nalog in risanju polinomov glede na njihov vnos, omogočal bi reševanje nalog na treh težavnostnih nivojih ter tudi reševanje preizkusov znanja, ki bi se lahko shranili na spletno stran.

Za dosego tega cilja sva uporabila različne metode raziskovanja, ki so nama omogočile pridobiti potrebno znanje in informacije za razvoj programa.

Ena od glavnih metod, ki sva jo uporabila pri raziskovanju, je bila anketa. Sestavila sva vprašalnik, ki je zajemal mnenja dijakov glede uporabe matematičnih programov za preverjanje in reševanje matematičnih nalog. Anketo sva nato poslala ostalim dijakom naše šole in na ta način zbrala veliko podatkov, ki so nama pomagali pri oblikovanju koncepta in funkcionalnosti programa.

Druga pomembna metoda raziskovanja, ki sva jo uporabila, je bila analiza dokumentov. Preučila sva različne publikacije, učbenike in druge vire, ki so obravnavali podobne teme, kot sva jih zajela v raziskovalni nalogi. V tem procesu sva pridobila veliko znanja in idej za oblikovanje in razvoj programa.

Poleg tega sva opravila tudi pregled virov, vključno s spletnimi forumi in drugimi viri, kjer sva pridobila dodatne informacije o ostalih matematičnih programih. To nama je pomagalo, da sva se osredotočila na najbolj pomembne funkcionalnosti in načine uporabe programa, ki bi bili najbolj koristni za uporabnike.

Tudi eksperimentiranje je bila pomembna metoda raziskovanja, ki sva jo uporabila pri razvoju programa. Izvedla sva številne teste in poskusne primere za preizkušanje in izboljšanje funkcionalnosti programa. S tem sva lahko ugotovila, kje so morebitne napake in težave ter jih odpravila, da bi program deloval čim bolj učinkovito.

Nazadnje sva opravila intervju s profesorico matematike, ki nama je pomagala pridobiti vpogled v to, kaj je za učitelje in dijake najpomembnejše pri uporabi matematičnih programov za preverjanje in reševanje matematičnih nalog. To nama je omogočilo, da sva se osredotočila na funkcionalnosti, ki so najbolj pomembne za uporabnike, in jih vključila v program.

2 Anketa

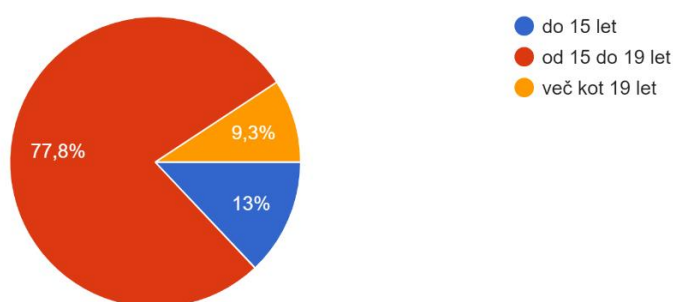
Namen te ankete je bil ugotoviti mnenje dijakov glede uporabnosti matematičnega programa, ki bi jim pomagal pri učenju in razumevanju matematičnih konceptov. Želela sva izvedeti, ali bi takšen program izboljšal njihovo uspešnost pri matematiki in kako bi lahko program oblikovala, da bi bil čim bolj uporaben za dijake. Poleg tega sva želela ugotoviti tudi, ali bi dijaki sploh bili pripravljeni uporabljati matematični program in kakšne funkcionalnosti bi si želeli vanj vključene. Želela sva pridobiti mnenje dijakov glede uporabnosti in potencialnih prednosti takega programa ter ugotoviti morebitne pomanjkljivosti, ki bi jih lahko izboljšala. Zato sva izvedla anketo med dijaki, da bi pridobila čim bolj celovit vpogled v njihove potrebe in želje v zvezi z uporabo matematičnega programa.

2.1 Izbor in struktura vzorca anketiranih

Raziskavo sva izvedla s pomočjo spletnega vprašalnika v Google Forms. Anketo je izpolnilo 54 dijakov. S to raziskavo sva želela ugotoviti, če se drugim dijakom zdi najina ideja za program primerna in uporabna.

Starost:

Koliko si star/a
54 odgovorov

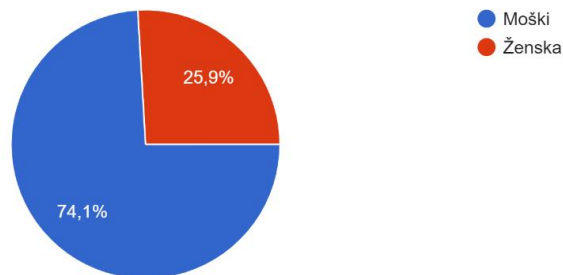


Slika 1: Starost anketirancev.

Anketo so večinoma reševali dijaki, stari od 15 do 19 let, kar je tudi najina ciljna skupina za uporabnike.

Spol:

Katerega spola si?
54 odgovorov

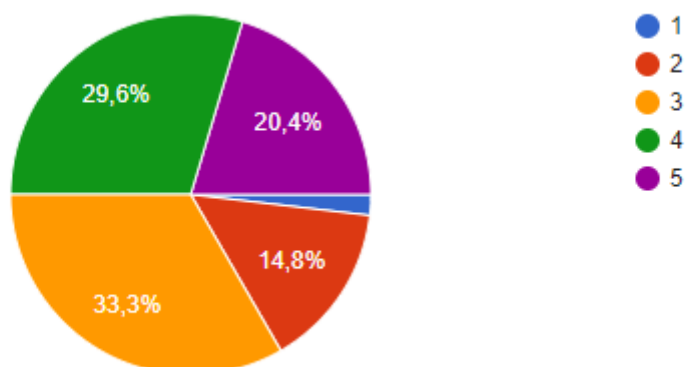


Slika 2: Spol anketirancev.

Med anketiranci je bilo skoraj tri četrtine moških.

2.2 Rezultati ankete

Kako dobro ti gre matematika (1-5)?

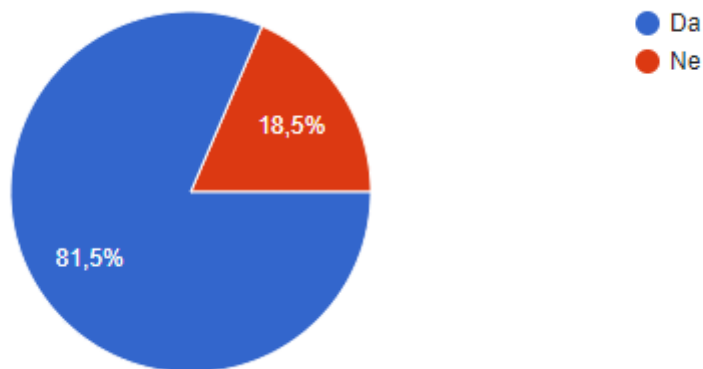


Slika 3: Uspeh anketirancev pri matematiki.

Največ anketirancev (33,3 odstotkov) je izjavilo, da imajo pri matematiki oceno 3. Veliko anketirancev je napisalo, da imajo pri matematiki oceno 4 in samo eden, da ima oceno 1.

Ali bi tak program redno uporabljal/a?

Iz ankete sva ugotovila, da veliko dijakov potrebuje matematični program, ki vsebuje vse na enem mestu, saj je kar 81 odstotkov izbralo opcijo »Da« pri tem vprašanju.



Slika 4: Mnenje o uporabi.

Iz ankete sva tudi ugotovila, da dijakom največ težav povzroča geometrija, zato sva se odločila izdelati program, poimenovan hitra matematika. Rezultati so pokazali tudi, da dijaki menijo, da bi se s pomočjo takšnega programa učili.

3 Primerjava z že obstoječima programoma

Med raziskovanjem matematičnih programov sva naletela na več aplikacij, ki imajo nekaj skupnih lastnosti z najinim programom, vendar kljub temu obstajajo razlike.

Eden izmed programov, ki je podoben najinemu je GeoGebra.

Najin program se osredotoča bolj na reševanje matematičnih nalog in izboljšanje znanja, medtem ko lahko s pomočjo GeoGebre konstruiramo točke, vektorje, daljice, premice, ... Prav tako lahko vse elemente dinamično spreminjamo. GeoGebra ima tudi podrobnejšo dokumentacijo in veliko skupnost uporabnikov.

Najin program ima vsekakor prednosti v smislu dodatnih funkcij, kot sta dvostopenjsko overjanje in reševanje vaj, ki niso na voljo v GeoGebri.

Ena izmed prednosti GeoGebre je njena priljubljenost in široka uporaba, kar pomeni, da je veliko virov za učenje in podporo, če kdo naleti na težave. Poleg tega ima tudi podrobno dokumentacijo in številne priročnike, ki so na voljo v različnih jezikih.

Na splošno bi bilo odvisno od uporabnikovih potreb in preference, kateri program bi bil boljši za njih. Najin matematični program je bolj primeren za tiste, ki želijo reševati matematične naloge in izboljšati svoje znanje, medtem ko je GeoGebra bolj primerna za tiste, ki želijo raziskovati in izdelovati geometrijske diagrame.

Pregledala sva tudi program Desmos, ki omogoča risanje grafov funkcij. Nudi veliko možnosti risanja funkcij z različnimi barvami in oblikami in ima možnost dodajanja oznak za lažjo preglednost. Omogoča tudi reševanje enačb in izračunavanje matematičnih izrazov. Vsebuje spletno učilnico, ki omogoča uporabnikom, da izvajajo interaktivne matematične simulacije in rešujejo kompleksne naloge, da lahko učitelji razlagajo snov. To funkcijo bi v prihodnje lahko dodala tudi v najin program.

Zgoraj predstavljena programa sta usmerjena bolj v risanje grafov funkcij, nimata pa možnosti za računanje z liki in telesi. Ti programi omogočajo risanje grafov funkcij in predstavitev formul za like in telesa na grafični način. Najin program pa sva zasnovala tako, da omogoča računanje z liki in telesi. Potrebno je povedati, da bova program redno posodabljala, kar pomeni, da bova lahko gradila na skupnosti in nadgrajevala varnost programa. Redne posodobitve prav tako pripomorejo k večji stabilnosti in zmogljivosti programa, saj lahko sproti izboljšujeva kodo in odpraviva morebitne napake.

4 Program

Program je namenjen tistim, ki želijo preveriti svoje znanje s področja računanja ploščin in ostalih podatkov likov, prostornin ter površine teles. Program je v pomoč tudi pri pretvarjanju enot ter risanju linearnih in kvadratnih funkcij, polinomov ter logaritemske funkcije tako, da se vnese njihove koeficiente. Ponuja tudi možnost naključnega izbiranja nalog za vajo, kar omogoča testiranje znanja. Na spletni strani je tudi možen ogled nalog, ki so bile predhodno rešene.

4.1 Visual Studio

4.1.1 Splošno

Visual Studio je integrirano razvojno okolje (IDE), ki ga je razvil Microsoft. Razvijalci ga uporabljajo za ustvarjanje programskih aplikacij za različne platforme, vključno z operacijskimi sistemi Windows, macOS, Linux, iOS, Android in spletnimi aplikacijami. Visual Studio ponuja vrsto funkcij, vključno z urejevalnikom kode, orodji za odpravljanje napak in orodji za upravljanje projektov.

4.1.2 Windows Forms

Windows Forms je nabor orodij uporabniškega vmesnika, ki ga zagotavlja Microsoft za ustvarjanje namiznih aplikacij v ogrodju .NET. Zagotavlja preprost način ustvarjanja uporabniških vmesnikov s funkcijo »povleci in spusti«, ki jih je mogoče prilagoditi z uporabo C# ali Visual Basic.NET. Windows Forms ponuja široko paleto vnaprej pripravljenih kontrol, kot so besedilna polja, gumbi in oznake, ki jih je mogoče preprosto dodati v aplikacijo.

Obstaja več razlogov, zakaj sva za razvoj matematičnega programa izbrala Visual Studio Windows Forms. Najprej nama je vmesnik »povleci in spusti« olajšal ustvarjanje uporabniku prijazne aplikacije brez obsežnih izkušenj s programiranjem. Drugič, program Windows Forms ponuja vrsto vnaprej pripravljenih kontrol, ki sva jih lahko uporabila za ustvarjanje prilagojenega vmesnika za aplikacijo. Prav tako sva lahko izbirala med različnimi programskimi jeziki, vključno s C# in Visual Basic.NET, kar nama je omogočilo, da sva program razvila po svojih željah.

V veliko pomoč nama je bila tudi velika skupnost razvijalcev, ki uporabljajo Visual Studio. Ustvarili in delili so knjižnice, ki sva jih zlahka vključila v najin program. To nama je prihranilo čas in nama omogočilo dodajanje različnih funkcij v aplikacijo, ne da bi kodo pisala od začetka.

Prav tako so bila orodja za razhroščevanje, ki jih ponuja Visual Studio uporabna pri odkrivanju in odpravljanju težav v najini kodi.

Vendar morava opozoriti, da je Windows Forms namenjen samo namiznim aplikacijam na Windows operacijskem sistemu, kar pomeni, da če bi se v prihodnosti odločila za razširitev aplikacije na druge operacijske sisteme, bi morala izbrati drugo razvojno okolje.

Če povzameva, sva Visual Studio Windows Forms izbrala zaradi njegovega intuitivnega vmesnika, vnaprej pripravljenih kontrol, prilagodljivosti jezika in velike skupnosti razvijalcev. Čeprav je omejen samo za razvoj Windows namiznih aplikacij, je zanesljivo in učinkovito razvojno okolje.

4.2 C#

4.2.1 Splošno

C# je univerzalni objektno orientiran programski jezik, ki ga uporabljajo programerji po vsem svetu. Uporabljen je za razvoj številnih aplikacij in iger. C# je bil razvit s strani Microsofta in se uporablja za pisanje programske kode, ki se izvaja na .NET Framework-u in .NET Core platformah. Tako lahko programerji ustvarijo aplikacije, ki delujejo na različnih napravah in operacijskih sistemih, kot so Windows, Linux in MacOS.

Programiranje v C# je podobno kot v drugih programskih jezikih, kot sta Java in C++. V primeru, da že imate izkušnje z enim od teh jezikov, bo to zelo olajšalo vaše delo s C#.

Poleg tega je C# zelo prilagodljiv in se lahko uporablja za razvoj različnih vrst aplikacij, vključno z namiznimi in spletnimi aplikacijami igrami. Ravno tako lahko razvijalci izpolnijo vse zahteve uporabnikov.

4.2.2 Zgodovina

C# je bil prvič predstavljen leta 2000. Predstavil ga je Andersa Hejlsberga iz podjetja Microsoft in ga je leta 2002 Ecma odobrila kot mednarodni standard, leto kasneje pa še ISO/IEC. Microsoft je naenkrat predstavil C#, .NET Framework in razvojno okolje Visual Studio. Štiri leta kasneje se je začel projekt Mono, ki je bil brezplačen in odprtokoden ter je omogočal prenosni prevajalnik in izvršno okolje za programski jezik C#. Deset let kasneje je Microsoft izdal Visual Studio Code (urejevalnik kode), Roslyn (prevajalnik) in združeno .NET platformo (programsko ogrodje).

Najnovejša stabilna različica je C# 11.0, ki je bila izdana leta 2022 v .NET 7.0.

4.3 Knjižnice

4.3.1 ZedGraph

ZedGraph je odprtokodna knjižnica grafov za aplikacije Microsoft .NET, ki razvijalcem omogoča enostavno dodajanje grafov in diagramov v aplikacije. Razvil jo je razvil John Champion s sodelavci. Pogosto se uporablja v različnih panogah, vključno s financami, inženiringom in znanstvenimi raziskavami.

Knjižnica podpira številne vrste grafov in napredne funkcije, avtomatsko prilagajanje številke na koordinatnem sistemu, povečevanje ter pomikanje, vključuje pa tudi vgrajeno podporo za interaktivne vmesnike. Ima popolno dokumentacijo, kar pomeni, da je to knjižnico enostavno uporabljati.

Njena prednost je tudi prilagodljivost, ki je zasnovana tako, da razvijalcem omogoča prilagajanje že obstoječih funkcij ZedGraph glede na njihove želje. Razširiti jo je mogoče tudi prek vmesnikov API (application programming interface), kar razvijalcem omogoča, da knjižnici dodajo lastne funkcije in jo povežejo z drugimi knjižnicami ter ogrodji.

Poleg zmožnosti izdelave grafov ponuja tudi visoko zmogljivost in zanesljivost, saj je dobro optimizirana za hitro delovanje in lahko obdeluje velike količine podatkov.

Knjižnica je licencirana pod licenco GNU Lesser General Public License (LGPL), ki omogoča komercialno uporabo, zato je idealna za razvijalce, ki morajo svojim aplikacijam dodati vizualizacijo podatkov.

To knjižnico sva uporabila za risanje grafov. Knjižnica omogoča programu več opcij za prilagajanje, kot je na primer povečevanje in premikanje grafa.

4.3.2 Newtonsoft.Json

V jeziku C# se pogosto uporablja priljubljena knjižnica za delo s podatki JSON, ki je knjižnica tretje osebe. Zagotavlja zmogljiv nabor funkcij za razčlenjevanje, generiranje in upravljanje podatkov JSON.

JSON (JavaScript Object Notation) je lahka oblika izmenjave podatkov, ki se pogosto uporablja za prenos podatkov med strežniki in odjemalci. Njegova sintaksa temelji na podnaboru jezika JavaScript, in ker je enostaven za branje in pisanje, se pogosto uporablja.

Ena od ključnih lastnosti knjižnice Newtonsoft.Json je njena zmožnost serializacije in deserializacije podatkov JSON. Serializacija je postopek pretvorbe objekta v niz JSON, medtem ko je deserializacija postopek pretvorbe niza JSON v objekt.

Recimo, da imamo razred z imenom *Naloga*, ki ima štiri lastnosti: ime naloge, kategorijo, stopnjo težavnosti in pravilni odgovor. Metodo `JsonConvert.SerializeObject` lahko uporabimo za serializacijo primera *Naloga* v niz JSON na naslednji način:

```
Naloga naloga = new Naloga
{
    imeNaloge = "Računanje višine trikotnika",
    kategorija = "Geometrija",
    stopnjaTežavnosti = 1,
    pravilniOdgovor = 4.9
};

string json = JsonConvert.SerializeObject(naloga);
```

Slika 5: Uporaba metode `JsonConvert.SerializeObject`.

Nastal bo niz JSON, ki bo izgledal tako:

```
{
  "imeNaloge": "Računanje višine trikotnika",
  "kategorija": "Geometrija",
  "stopnjaTežavnosti": 1,
  "pravilniOdgovor" = 4.9
}
```

Slika 6: Niz JSON

Če povzamemo, je *Newtonsoft.Json* zmogljiva in prilagodljiva knjižnica, ki zagotavlja številne funkcije za delo s podatki JSON v jeziku C#. V aplikaciji *Windows Forms* je lahko uporabljena za serializacijo in deserializacijo podatkov, poizvedovanje in manipulacijo podatkov z uporabo LINQ ter prilagajanje postopka serializacije in deserializacije. S svojim bogatim naborom funkcij in enostavnim API-jem je priljubljena izbira za razvijalce C#, ki morajo delati s podatki JSON.

V najinem primeru sva to knjižnico uporabila za razčlenitev odgovora JSON iz API. To nama je omogočilo, da sva iz odgovora izluščila ustrezne informacije in jih uporabila v najinem programu. Knjižnica zagotavlja preprost in učinkovit način dela s podatki JSON v .NET.

4.3.3 ZXing

ZXing je odprtokodna knjižnica, ki se pogosto uporablja za dekodiranje in kodiranje črtnih kod in QR kod. Knjižnico je leta 2007 prvič razvil Sean Owen kot knjižnico Java, pozneje pa je bila prenesena v druge programske jezike, kot je C#. Znana je po hitrem in natančnem dekodiranju črtnih in QR kod ter je združljiva s številnimi operacijskimi sistemi in platformami.

Ena najpomembnejših prednosti knjižnice ZXing je njena podpora širokemu naboru formatov črtnih in QR kod. Dekodira in kodira več formatov kod, vključno z UPC, EAN, Code 39, Code 128 in številnimi drugimi. Zaradi tega je to vsestranska knjižnica, ki se lahko uporablja v različnih aplikacijah.

Druga značilnost, zaradi katere ZXing izstopa, je njena enostavna uporaba. Zagotavlja preprost vmesnik API, ki omogoča enostavno vključitev funkcionalnosti skeniranja in generiranja črtnih in QR kod v aplikacijo. To pomeni, da lahko razvijalci knjižnico hitro in enostavno vključijo v svoje projekte, ne da bi jim bilo treba porabiti veliko časa za nastavitve in konfiguracijo.

Knjižnica je tudi zelo prilagodljiva, saj zagotavlja več možnosti za natančno prilagoditev postopka skeniranja in generiranja. Razvijalci lahko na primer določijo format kode, ki ga želijo dekodirati ali kodirati, prilagodijo ločljivost skeniranja ter omogočijo ali onemogočijo določene vrste dekodiranja.

Ena od njenih ključnih prednosti je zmožnost obdelave poškodovanih ali delno zakritih kod, saj uporablja izpopolnjene algoritme za analizo slik kod ter določi pravilno obliko, tudi če je koda nepopolna ali poškodovana.

Združljiva je s številnimi operacijskimi sistemi in platformami, vključno z operacijskimi sistemi Windows, Mac, Linux, iOS in Android. To pomeni, da lahko razvijalci knjižnico uporabljajo v različnih aplikacijah, od namiznih do mobilnih aplikacij.

Podpira tudi več programskih jezikov, vključno z jeziki Java, C#, C++, Python in Ruby, kar pomeni, da se lahko uporablja v zelenem programskem jeziku.

Na splošno je ZXing zmogljiva in vsestranska knjižnica, ki se lahko uporablja v številnih aplikacijah. Zaradi podpore za več programskih jezikov, širokega nabora formatov črtnih in QR kod ter preproste uporabe je odlična izbira za razvijalce, ki morajo delati s črtnimi ali QR kodami.

Knjižnico ZXing sva uporabila za kreiranje QR kode, ki se ustvari s pomočjo skrivnega ključa. Uporabnik mora na svojo mobilno napravo najprej namestiti aplikacijo Google Authenticator, ki mu omogoča skeniranje QR kode. Po uspešnem skeniranju se bo vsakih trideset sekund ustvarila nova šestmestna številka, ki jo bo uporabnik vpisal ob prijavi v najin program. To bo uporabniku omogočilo uspešno in varno uporabo aplikacije.

4.3.4 Otp.NET

Otp.NET je knjižnica za ustvarjanje in potrjevanje enkratnih gesel (OTP – one time password) v aplikacijah .NET. OTP je vrsta avtentikacijske kode, ki se uporablja za zagotavljanje dodatne stopnje varnosti pri prijavi v aplikacijo ali na spletno mesto. Zagotavlja preprost in za uporabo enostaven API (application programming interface) za generiranje in potrjevanje OTP, zato je odlična izbira za razvijalce, ki morajo v svoje aplikacije .NET uvesti dvostopenjsko preverjanje pristnosti.

Ena od ključnih prednosti je njena podpora za več algoritmov OTP. Podpira več algoritmov, vključno z algoritmom enkratnega gesla na podlagi časa (TOTP) in algoritmom enkratnega gesla na podlagi HMAC (HOTP). To razvijalcem omogoča, da izberejo algoritem, ki najbolj ustreza njihovim potrebam in potrebam njihovih uporabnikov.

Je tudi zelo prilagodljiva, saj zagotavlja več možnosti za natančno prilagoditev postopka ustvarjanja in potrjevanja OTP. Razvijalci lahko na primer določijo dolžino enkratnega gesla, časovno okno za kode TOTP in število števk v geslu. Ta prilagodljivost omogoča razvijalcem, da postopek ustvarjanja in potrjevanja OTP prilagodijo svojemu posebnemu primeru uporabe.

Druga značilnost, ki razlikuje Otp.NET od drugih knjižnic OTP, je enostavnost uporabe. Zagotavlja preprost API, ki omogoča enostavno vključitev funkcionalnosti OTP v aplikacijo. Za generiranje enkratnega gesla razvijalci preprosto uporabijo metodo GenerateOTP in posredujejo skrivni ključ uporabnika. Za potrditev enkratnega gesla razvijalci uporabijo metodo ValidateOTP ter posredujejo uporabnikov skrivni ključ in vrednost gesla. Knjižnica poskrbi za vse ostalo, kar olajša izvajanje funkcionalnosti OTP, ne da bi bilo treba porabiti veliko časa za nastavitve in konfiguracijo.

Vključuje tudi več dodatnih funkcij, ki so lahko koristne za razvijalce. Knjižnica na primer zagotavlja objekt TimeWindow, ki ga je mogoče uporabiti za izračun trenutnega časovnega okna za kode TOTP. Vsebuje tudi objekt OTPUrl, ki ga je mogoče uporabiti za generiranje

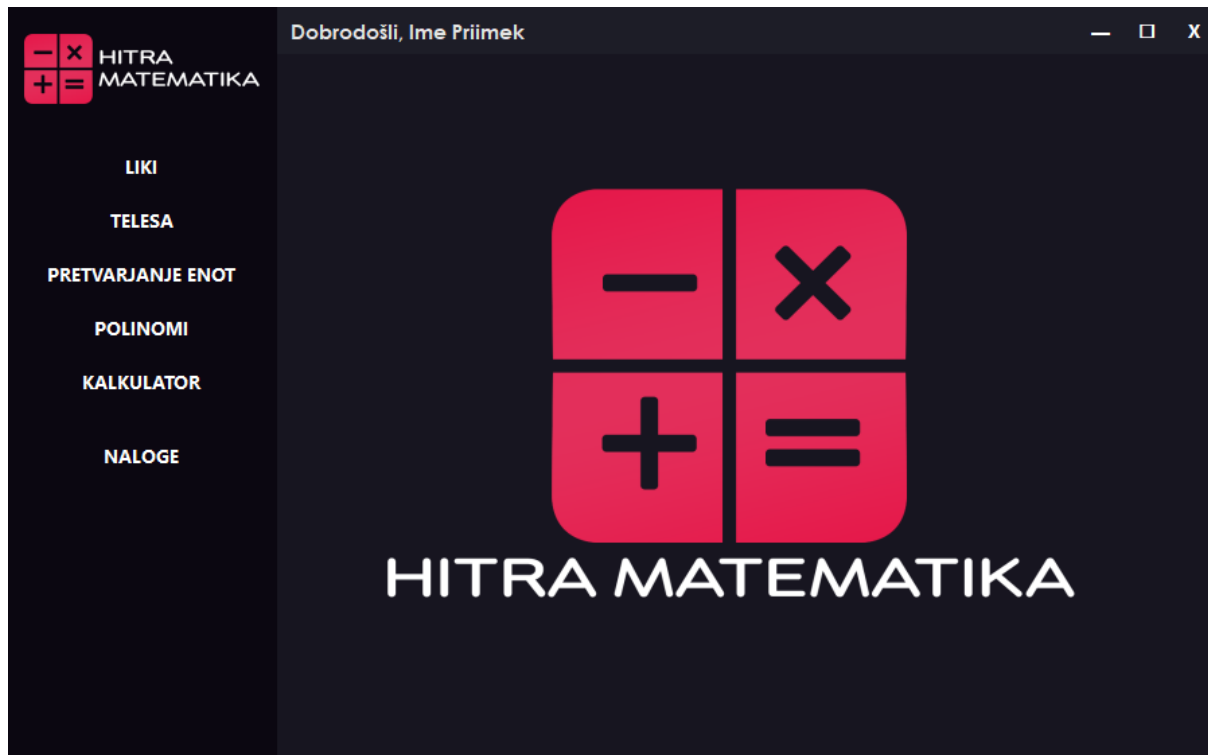
naslova URL, ki ga lahko skenira aplikacija OTP, kar olajša integracijo funkcionalnosti enkratnih gesel v mobilno aplikacijo ali drugo zunanjo aplikacijo.

Na splošno je OTP.NET zmogljiva in vsestranska knjižnica, ki se lahko uporablja za izvajanje funkcionalnosti enkratnih gesel v aplikacijah .NET. Zaradi podpore za več algoritmov OTP, možnosti prilagajanja in enostavne uporabe je odlična izbira za razvijalce, ki morajo v svoje aplikacije uvesti dvostopenjsko overjanje.

Knjižnico sva uporabila za generiranje TOTP kode na podlagi skrivnega ključa, ki se ob registraciji najprej naključno ustvari, potem se ključ šifrira z ustrezno metodo, ko pa uporabnik ob prijavi uporabi dvostopenjsko overjanje, se ta skrivni ključ pokliče in dešifrira, kar omogoča kreiranje unikatne QR kode.

4.4 Delovanje programa

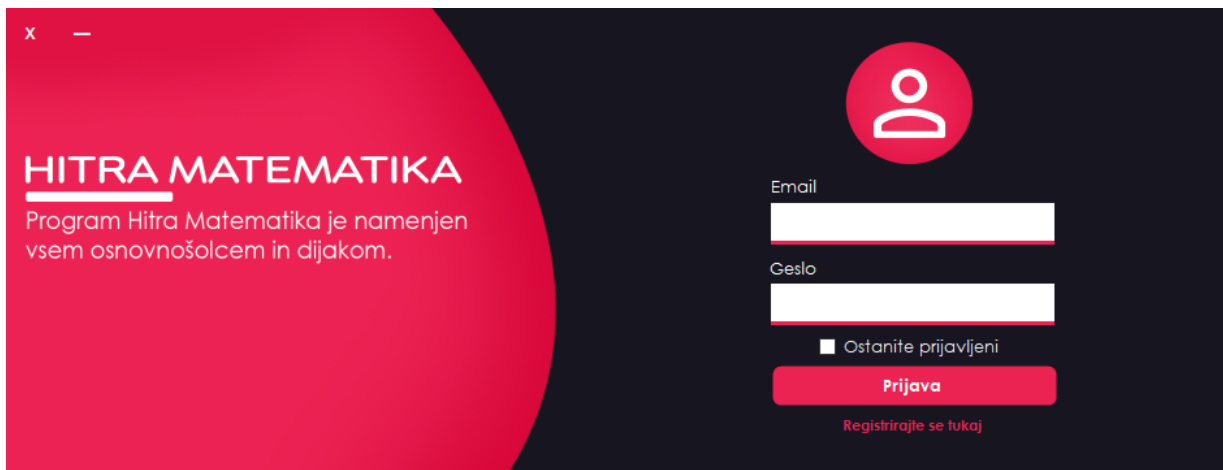
Ko odpremo program, se moramo najprej registrirati na spletni strani. Če je uporabnik že registriran, se lahko prijavi v program. Nato ima uporabnik na levi strani meni, v katerem lahko izbira med računanjem z liki in telesi, pretvarjanjem enot, risanjem grafov, nalogami ter preverjanjem znanj.



Slika 7: Program

4.4.1 Okno za prijavo

Ob uporabnikovem vpisu podatkov za prijavo se obrazec podatkov pošlje na PHP REST API, ki jih preveri v podatkovni bazi. V primeru, da se uporabniško ime in geslo ujemata, se pošlje nazaj odgovor "success" v obliki JSON skupaj z dodatnimi podatki o uporabniku, kot sta ime in priimek. V nasprotnem primeru se pošlje nazaj sporočilo o napaki.



Slika 8: Okno za prijavo.

Ob kliku na gumb »prijava« program prebere uporabniško ime in geslo, ki ga je uporabnik vnesel v tekstovna polja. Nato se podatki pošljejo s pomočjo HTTP POST zahtevka na URL naslov PHP REST API¹, ki se nahaja na najini spletni strani. Za to uporablja HttpClient razred. Nato se pošlje obrazec podatkov, ki vsebuje e-naslov in geslo. V primeru, da je bila zahteva uspešna, se odgovor prebere in pretvori v JSON. Če je prijava uspela, se prikaže glavno okno aplikacije Windows Forms, ki je skrito, dokler se uporabnik ne prijavi. V primeru, da prijava ni uspela, se prikaže sporočilo o napaki.

```
try
{
    var response = await client.PostAsync(url, vsebina);
    var responseString = await response.Content.ReadAsStringAsync();
    var responseObject = JObject.Parse(responseString);
    if (responseObject["status"].ToString() == "success")
    {
        Form_glavna.glavnoOkno = new Form_glavna();
        glavnoOkno.ImeUporabnika = responseObject["ime"].ToString();
        glavnoOkno.PriimekUporabnika = responseObject["priimek"].ToString();
        glavnoOkno.Show();
        ShraniPodatke();
        this.Hide();
    }
    else
    {
        MessageBox.Show(responseObject["message"].ToString(), "Napaka pri prijavi", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
catch (Exception ex)
{
    MessageBox.Show("Prišlo je do napake pri prijavi. Poskusite ponovno kasneje.", "Napaka pri prijavi", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
```

Slika 9: Del kode za vpis v program.

¹ Programski vmesnik aplikacij.

Uporabnik ima tudi možnost obkljukati potrditveno polje, ki je namenjeno temu, da se podatki ob uspešni prijavi shranijo v lokalno podatkovno bazo. Ko uporabnik znova zažene program, se preveri, ali so podatki o uporabniku shranjeni v lokalni bazi podatkov. V primeru, da so, se e-naslov in geslo samodejno vpišeta v tekstovna polja, prav tako pa se označi potrditveno polje »ostanite prijavljeni«.

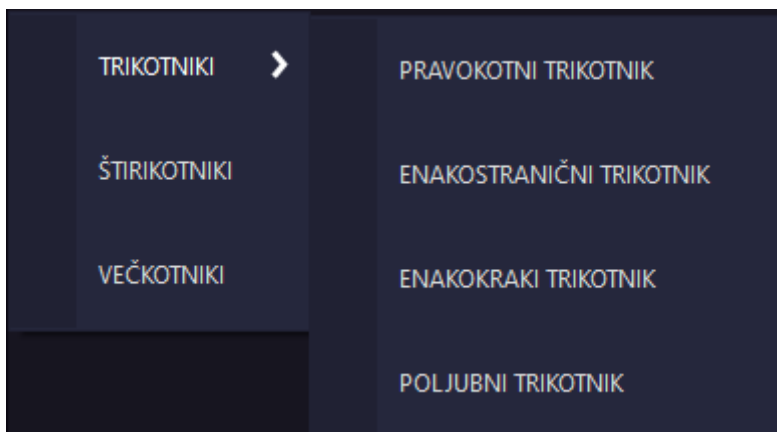
```
void ShraniPodatke()
{
    if (chk_OstaniPrijavljen.Checked == true)
    {
        Properties.Settings.Default.email = txtEmail.Text;
        Properties.Settings.Default.geslo = txtGeslo.Text;
        Properties.Settings.Default.Save();
    }
    else
    {
        Properties.Settings.Default.email = "";
        Properties.Settings.Default.geslo = "";
        Properties.Settings.Default.Save();
    }
}

void NaloziPodatke()
{
    if (Properties.Settings.Default.email != string.Empty)
    {
        txtEmail.Text = Properties.Settings.Default.email;
        txtGeslo.Text = Properties.Settings.Default.geslo;
        chk_OstaniPrijavljen.Checked = true;
    }
}
```

Slika 10: Shranjevanje podatkov v lokalno podatkovno bazo.

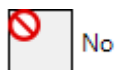
4.4.2 Liki

V primeru, da v meniju izberemo like, se nam odpre podmeni, v katerem lahko izbiramo med trikotniki, štirikotniki in večkotniki. Ko izberemo želen tip lika, se nam odpre še en podmeni, v katerem lahko izberemo ustrezen lik, na primer pravokotni trikotnik.



Slika 11: Podmeni.

Tako se nam odpre okno, ki omogoča vpis podatkov za želeni lik. Polja podatkov, ki jih ne moremo vnesti, so obarvana rdeče in so samo bralna. Ko gremo s kazalcem čez polje, se nam spremeni v znak za prepovedano.



No

Slika 12: Znak za prepovedano.

Program je zastavljen tako, da se polja, s katerimi v tem trenutku ne moremo izračunati vseh ostalih podatkov, sproti nastavijo na samo bralne (Slika 13). Na primer, če pri poljubnem trikotniku vpišemo dva kota, se bo tretji obarval rdeče, saj je iz treh kotov nemogoče izračunati dolžine stranic poljubnega trikotnika.

RAČUNANJE
Vnesi vstrezne podatke

| | | | |
|-------------|----------------------|--------------|---------------------------------|
| Stranica A: | <input type="text"/> | Alfa: | <input type="text" value="50"/> |
| Stranica B: | <input type="text"/> | Beta: | <input type="text" value="50"/> |
| Stranica C: | <input type="text"/> | Gama: | <input type="text"/> |
| | | Obseg: | <input type="text"/> |
| Ploščina: | <input type="text"/> | Višina na A: | <input type="text"/> |
| Polmer v: | <input type="text"/> | Višina na B: | <input type="text"/> |
| Polmer o: | <input type="text"/> | Višina na C: | <input type="text"/> |

Slika 13: Okno za poljubni trikotnik.

Za vsak lik sva si najprej naredila načrt, iz katerih podatkov je mogoče in smiselno izračunati ostale podatke. Nato sva naredila niz polj, uporabljenih v tem oknu z namenom, da bi bila koda v nadaljevanju krajša. Tako se ob vsaki spremembi v kateremkoli polju izvede metoda, ki preveri ustreznost podatkov.

```

TextBox[] textBoxes = new TextBox[] { txt_PravokotniTrikotnikStrA, txt_PravokotniTrikotnikStrB, txt_PravokotniTrikotnikStrC, txt_PravokotniTrikotnikStrA1 };
btn_PravokotniTrikotnikIzracunaj.Enabled = false;
if (txt_PravokotniTrikotnikStrA.Text != "" && txt_PravokotniTrikotnikStrB.Text != "" && txt_PravokotniTrikotnikStrC.Text != "" && txt_PravokotniTrikotnikStrA1.Text != "")
{
    foreach (TextBox textBox in textBoxes)
    {
        textBox.ReadOnly = true;
    }
}
else if (txt_PravokotniTrikotnikStrA.Text != "")
{
    if (txt_PravokotniTrikotnikStrB.Text != "")
    {
        foreach (TextBox textBox in textBoxes)
        {
            if (txt_PravokotniTrikotnikStrA != textBox && txt_PravokotniTrikotnikStrB != textBox)
            {
                textBox.ReadOnly = true;
            }
        }
        btn_PravokotniTrikotnikIzracunaj.Enabled = true;
    }
    else if (txt_PravokotniTrikotnikStrC.Text != "")
    {
        foreach (TextBox textBox in textBoxes)
        {
            if (txt_PravokotniTrikotnikStrA != textBox && txt_PravokotniTrikotnikStrC != textBox)
            {
                textBox.ReadOnly = true;
            }
        }
        btn_PravokotniTrikotnikIzracunaj.Enabled = true;
    }
    else if (txt_PravokotniTrikotnikStrA1.Text != "")
    {
        foreach (TextBox textBox in textBoxes)
        {
            if (txt_PravokotniTrikotnikStrA != textBox && txt_PravokotniTrikotnikStrA1 != textBox)
            {
                textBox.ReadOnly = true;
            }
        }
        btn_PravokotniTrikotnikIzracunaj.Enabled = true;
    }
}

```

Slika 14: Izsek kode za preverjanje podatkov pravokotnega trikotnika.

Program najprej preveri, če so vpisani vsi podatki in če so, nastavi vsa polja na samo bralna. To je mogoče samo ob izračunu podatkov, saj bi drugače ob vpisu več podatkov, kot je potrebnih, lahko prišlo do njihovih neskladnosti. Nato program preverja vsako možno kombinacijo podatkov. Najprej nastavi podatke na samo bralne, na koncu se pa nastavita še barva in kazalec, ki se pojavita ko imamo kazalec na tekstovnem polju.

```
foreach (TextBox textBox in textBoxes)
{
    if(textBox.ReadOnly == true)
    {
        textBox.BackColor = Color.FromArgb(255, 230, 28, 77);
        textBox.Cursor = Cursors.No;
        textBox.ForeColor = Color.FromArgb(255, 255, 255, 255);
    }
    else
    {
        textBox.BackColor = Color.White;
        textBox.Cursor = Cursors.IBeam;
        textBox.ForeColor = Color.FromArgb(255, 0, 0, 0);
    }
}
```

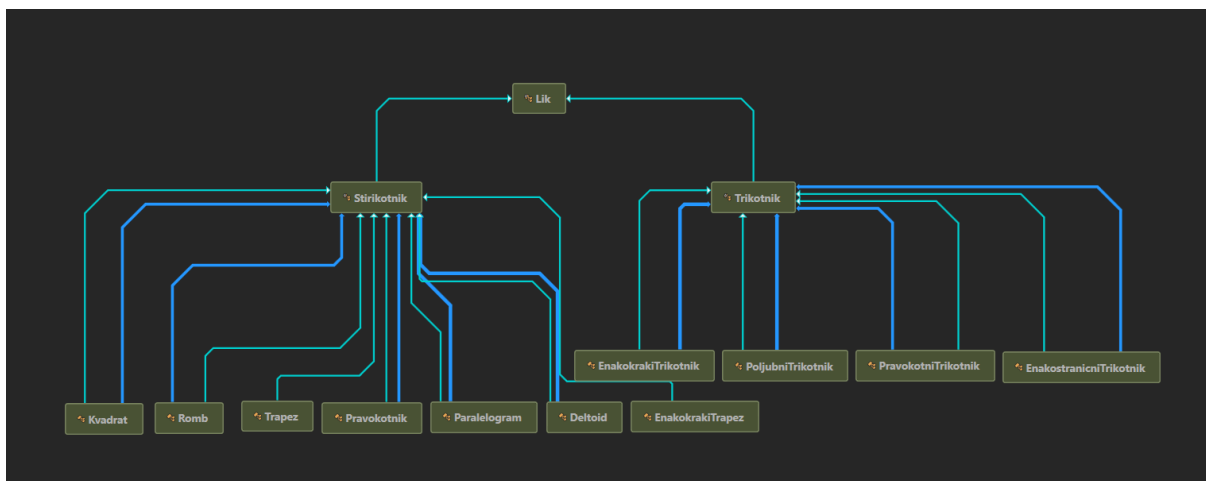
Slika 15: Izsek kazalca in barvo polj.

Gumb »Izračunaj« se omogoči, ko so vpisani vsi potrebni podatki. Ko pritisnemo na ta gumb, se najprej vsi podatki poskusijo pretvoriti v tip `double`, če to ne uspe, pa se vrednosti podatkov nastavijo na 0.

```
double.TryParse(txt_PravokotniTrikotnikStrA.Text, out double a);  
double.TryParse(txt_PravokotniTrikotnikStrB.Text, out double b);  
double.TryParse(txt_PravokotniTrikotnikStrC.Text, out double c);  
double.TryParse(txt_PravokotniTrikotnikStrA1.Text, out double a1);  
double.TryParse(txt_PravokotniTrikotnikStrB1.Text, out double b1);  
double.TryParse(txt_PravokotniTrikotnikVisC.Text, out double vc);  
double.TryParse(txt_PravokotniTrikotnikAlfa.Text, out double alfa);  
double.TryParse(txt_PravokotniTrikotnikBeta.Text, out double beta);
```

Slika 16: Pridobivanje podatkov iz polj.

Program z »if« stavki preveri, katere podatke je uporabnik vpisal in iz njih izračuna osnovne podatke (pri pravokotnem trikotniku so to stranice a, b in c) ter pokliče konstruktorja lika. Za računanje podatkov sva ustvarila svojo knjižnico. Najprej sva ustvarila abstraktni razred Liki, ki vsebuje dve lastnosti: obseg in ploščino. Nato sva ustvarila razred Trikotniki, ki deduje od razreda Liki in ima poleg lastnosti iz nadrazreda Liki še podatke za stranice a, b in c. Vsebuje pa tudi metodo, ki preverja, če trikotnik z vpisanimi podatki sploh obstaja, saj preveri, če je dolžina dveh stranic daljša od tretje. Iz razreda Trikotniki se nato dedujejo še ostali trikotniki. Vsak trikotnik ima konstruktorja, ki prejme podatke, iz katerih je možno izračunati vse ostale podatke, vsebuje pa tudi lastnosti, ki vračajo podatke o trikotniku.



Slika 17: Drevesna struktura razredov.

```
public class PravokotniTrikotnik : Trikotnik
{
    1 reference
    public PravokotniTrikotnik(double k1, double k2, double h)
    {
        if (preveriTrikotnik(k1, k2, h))
        {
            a = k1;
            b = k2;
            c = h;
        }
        else throw new Exception("To ni trikotnik");
    }

    2 references
    public double H
    {
        get
        {
            return Math.Round(c, 2);
        }
    }

    1 reference
    public double K1
    {
        get
        {
            return Math.Round(a, 2);
        }
    }

    1 reference
    public double K2
    {
        get
        {
            return Math.Round(b, 2);
        }
    }

    3 references
    public override double Obseg
    {
        get
        {
            return Math.Round(c + a + b, 2);
        }
    }
}
```

Slika 18: Razred PravokotniTrikotnik.

Tako se ob kliku na gumb »Izračunaj« ustvari primerek lika. Iz njegovih lastnosti nato dobimo še ostale podatke, ki so zaokroženi na dve decimaliki.

```
try
{
    t = new PravokotniTrikotnik(a, b, c);
    txt_PravokotniTrikotnikStrA.Text = t.K1.ToString();
    txt_PravokotniTrikotnikStrB.Text = t.K2.ToString();
    txt_PravokotniTrikotnikStrC.Text = t.H.ToString();
    txt_PravokotniTrikotnikStrA1.Text = t.A1.ToString();
    txt_PravokotniTrikotnikStrB1.Text = t.B1.ToString();
    txt_PravokotniTrikotnikVisC.Text = t.VisinaC.ToString();
    txt_PravokotniTrikotnikAlfa.Text = t.Alfa.ToString();
    txt_PravokotniTrikotnikBeta.Text = t.Beta.ToString();
    txt_PravokotniTrikotnikObseg.Text = t.Obseg.ToString();
    txt_PravokotniTrikotnikPloscina.Text = t.Ploscina.ToString();
}
```

Slika 19: Kreiranje pravokotnega trikotnika.

Na enak način, vendar z drugimi formulami in podatki, so narejeni tudi ostali tri- in štirikotniki. Pri pravilnih večkotnikih sva morala uporabiti statično metodo, ki lahko prejme različne kombinacije podatkov, metoda pa vrne vse ostale podatke.

```
public static double[] Izračunaj(double a, double n, double S, double o)
{
    double kot = 0;
    double notranjiKoti = 0;

    if (a > 0 && n > 0)
    {
        S = (n * Math.Pow(a, 2)) / (4 * Math.Tan(Math.PI / n));
        o = n * a;
    }
    else if (a > 0 && o > 0)
    {
        n = o / a;
        S = (n * Math.Pow(a, 2)) / (4 * Math.Tan(Math.PI / n));
    }
    else if (n > 0 && S > 0)
    {
        a = Math.Sqrt(S * 4 * Math.Tan(Math.PI / n) / n);
        o = n * a;
    }
    else if (n > 0 && o > 0)
    {
        a = o / n;
        S = n * Math.Pow(a, 2) * Math.Atan(Math.PI / n) / 4;
    }
    else return new double[] { };
    notranjiKoti = (n - 2) * 180.0;
    kot = notranjiKoti / n;
    return new double[] { a, n, S, o, kot, notranjiKoti };
}
```

Slika 20: Izračun večkotnika.

Vsak lik ima tudi razlago z vso potrebno snovjo in uporabljene formule.

4.4.3 Telesa

Da pridemo do opcije računanja s telesi, moramo v meniju izbrati telesa. Za računanje podatkov o telesih sva uporabila enak princip kot pri računanju z liki. Za računanje površine in prostornine sva uporabnikom omogočila vnos osnovne ploskve, višine in po potrebi dolžine stranic. Osnovno ploskev lahko uporabnik izračuna pri likih. Tako kot liki bodo tudi telesa imela vse potrebne razlage.

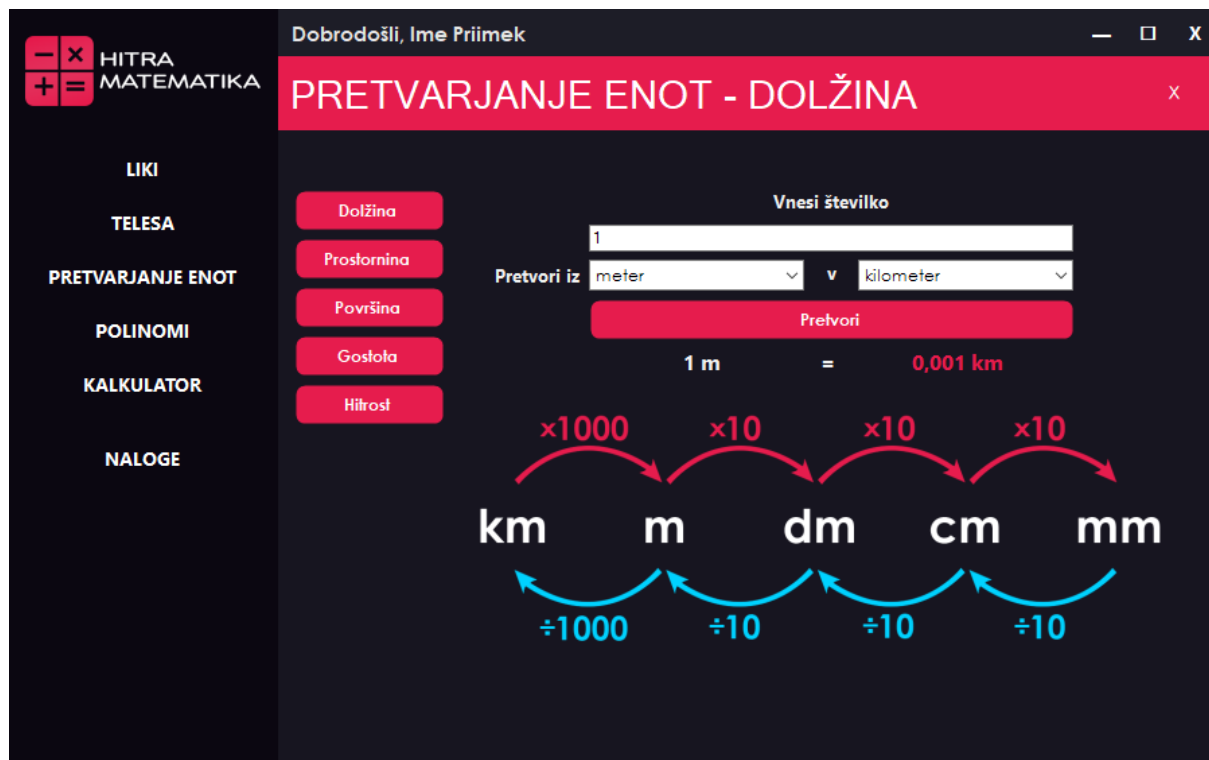
4.4.4 Uporabljene formule

V programu sva uporabila vrsto različnih formul za izračun podatkov. Največkrat sva uporabila naslednje formule:

- Pitagorov izrek,
- trigonometrijske funkcije,
- Heronovo formulo,
- višinski izrek in
- Evklidov izrek.

4.4.5 Pretvarjanje enot

V program sva dodala tudi možnost pretvarjanja enot, če bi jo kdo potreboval. Omogoča pretvarjanje enot dolžine, prostornine, površine, gostote in hitrosti.



Slika 21: Okno za pretvarjanje enot.

Zgornja slika prikazuje okno za pretvarjanje enot. V levem spustnem seznamu lahko izberemo enoto, iz katere želimo pretvarjati, v desnem pa, v katero enoto jo želimo pretvoriti. Za računanje sva uporabila samo statične bralne podatke, ki shranjujejo vrednosti enot. Na primer en meter je 0,001 kilometra.

Za uporabnikovo lažjo prestavo sva dodala še sliko, ki prikazuje razmerja med osnovnimi pretvorbami.

4.4.6 Risanje funkcij

Odločila sva se, da bo program omogočal risanje linearne in kvadratne funkcije, polinomov ter logaritemske funkcije.

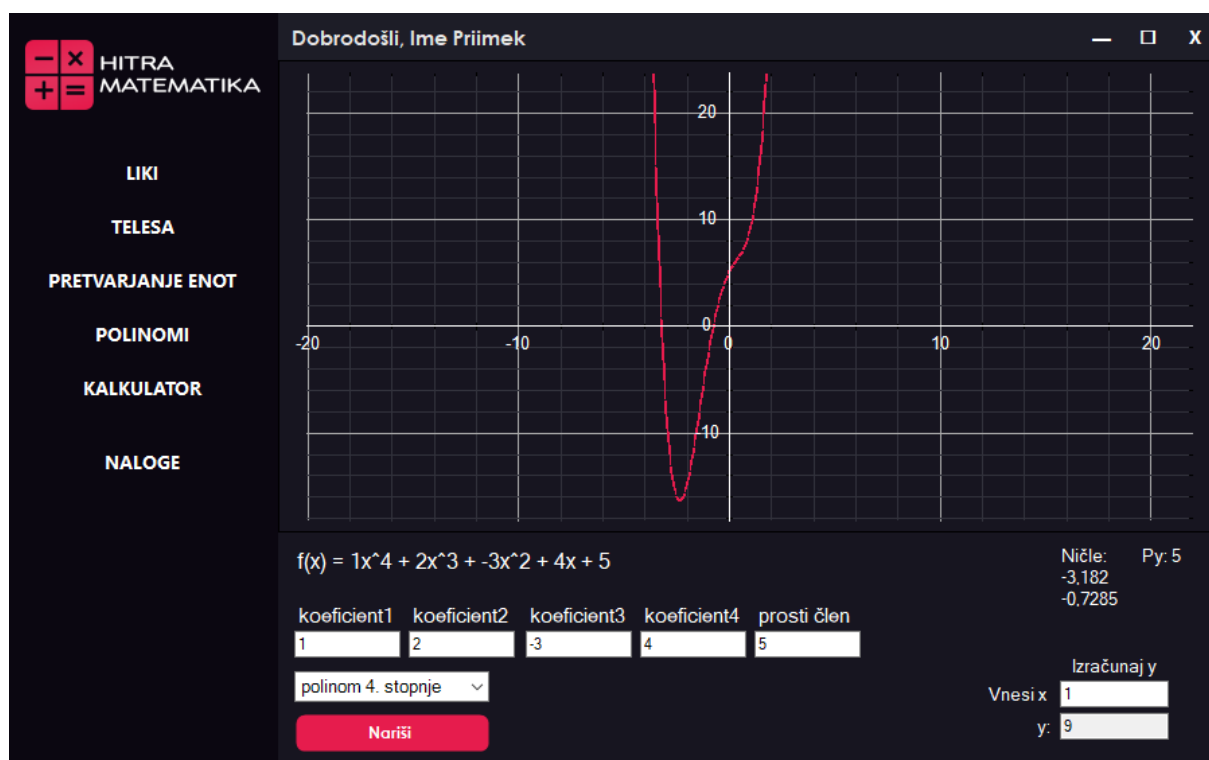
Za prikazovanje grafov sva uporabila knjižnico ZedGraph, saj omogoča veliko več možnosti kot graf, ki ga lahko najdemo v toolbox-u brez dodatnih knjižnic.

Program omogoča uporabniku izbor funkcije v spustnem seznamu in nastavljanje njenih koeficientov. Izbira lahko med linearno in kvadratno funkcijo, polinomi ter logaritemsko funkcijo. Število polj se sproti spreminja glede na izbran element v spustnem seznamu. Pod grafom pa lahko najdemo tudi zapis trenutne funkcije, kar omogoča boljšo preglednost.

$$f(x) = 1x^4 + 2x^3 + 3x^2 + 4x + 5$$

Slika 22: Prikaz funkcije.

Ob pritisku na gumb »Nariši« program pokliče metodo za risanje, ki pogleda, katera funkcija je izbrana v spustnem seznamu in glede na to uporabi ustrezno enačbo. Nato se izračunajo vse točke od -1000 pa do 1000 z intervalom 0.5. Poskusila sva tudi manjše intervale za bolj natančen graf, ampak sva ugotovila, da bi to preveč obremenjevalo uporabnikov računalnik. Točke se s pomočjo knjižnice ZedGraph² tudi samodejno povežejo in narišejo graf.



Slika 23: Okno za risanje grafov.

² Knjižnica za risanje grafov.

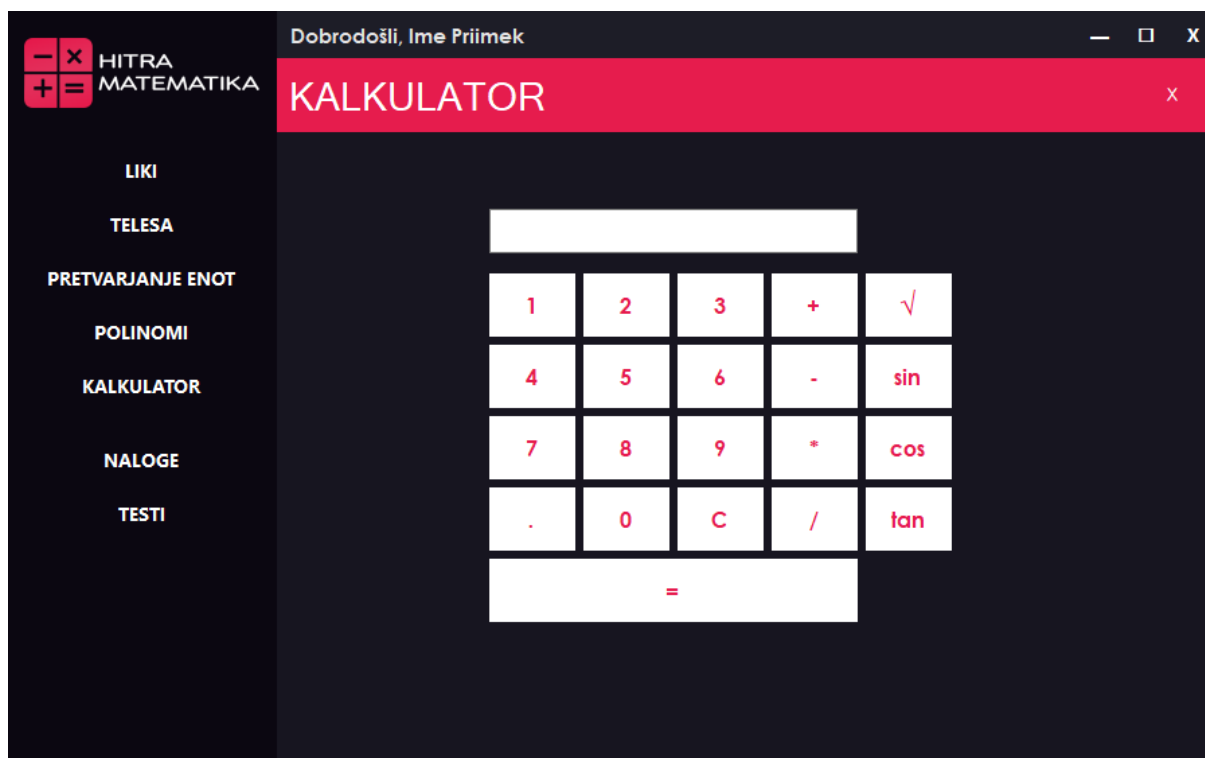
Kot lahko vidimo, program prikaže tudi ničle in presečišče z ordinatno osjo funkcije (Slika 23). Uporabnik ima tudi možnost vnosa vrednosti x funkcije, program pa mu izračuna vrednost y . Za računanje točk na grafu sva uporabila funkcijo z lambda operatorjem. Tako lahko program pokliče želeno funkcijo in iz nje dobi vrednost y na grafu.

```
Func<double, double> linearnaFunkcija = z => z * a + b;  
Func<double, double> kvadratnaFunkcija = z => Math.Pow(z, 2) * a + z * b + c;  
Func<double, double> polinomTretjeStopnje = z => Math.Pow(z, 3) * a + Math.Pow(z, 2) * b + z * c + d;  
Func<double, double> polinomCetrteStopnje = z => Math.Pow(z, 4) * a + Math.Pow(z, 3) * b + Math.Pow(z, 2) * c + z * d + e;  
Func<double, double> polinomPeteStopnje = z => Math.Pow(z, 5) * a + Math.Pow(z, 4) * b + Math.Pow(z, 3) * c + z * d + e + f;  
Func<double, double> logaritmi = z => a + Math.Log(z, b);
```

Slika 24: Funkcije za računanje točk.

4.4.7 Kalkulator

V program sva vključila tudi kalkulator (Slika 25), ki vključuje osnovne matematične operacije (seštevanje, odštevanje, množenje in deljenje) in nekaj funkcij, kot so koren, sinus, cosinus in tangens. Kalkulator vsebuje številčno tipkovnico, ki omogoča vnašanje števil v tekstovno polje. Uporabnik izbere operacijo, kalkulator pa izvede izbrano operacijo nad določenimi števili. Rezultat se prikaže v tekstovnem polju. Poleg osnovnih matematičnih operacij in funkcij ima kalkulator tudi gumbe za brisanje in vstavljanje decimalne vejice.

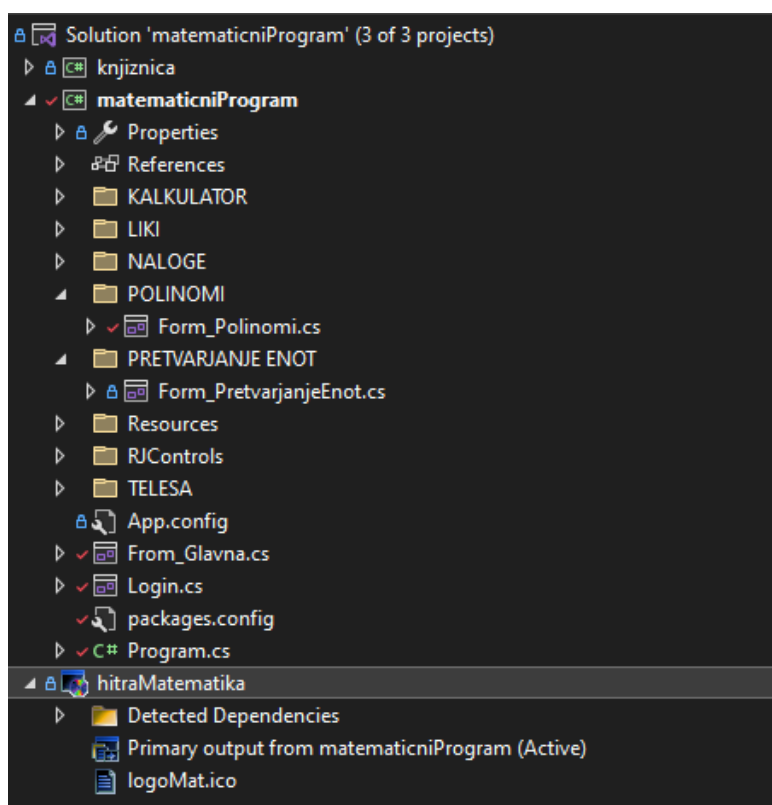


Slika 25: Okno kalkulator.

4.5 Namestitev

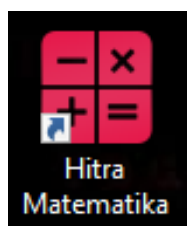
V projekt sva namestila razširitev »Microsoft Visual Studio Installer Projects 2022«, ki omogoča kreiranje instalcijskih datotek.

Nato sva v projekt dodala tudi ikono programa, ki je namenjena temu, da lahko uporabnik zazna in najde program na svojem računalniku. Na koncu sva namestila datoteko in preizkusila nameščen program. Preverila sva tudi, če se ikona programa pravilno prikaže v opravljeni vrstici.



Slika 26: Razširitev "Setup Project" za namestitev programa.

Pomembno je, da bova program redno posodabljala, saj bova s tem uporabnikom zagotovljala brehibno in učinkovito izkušnjo, ki jim bo olajšala delo.

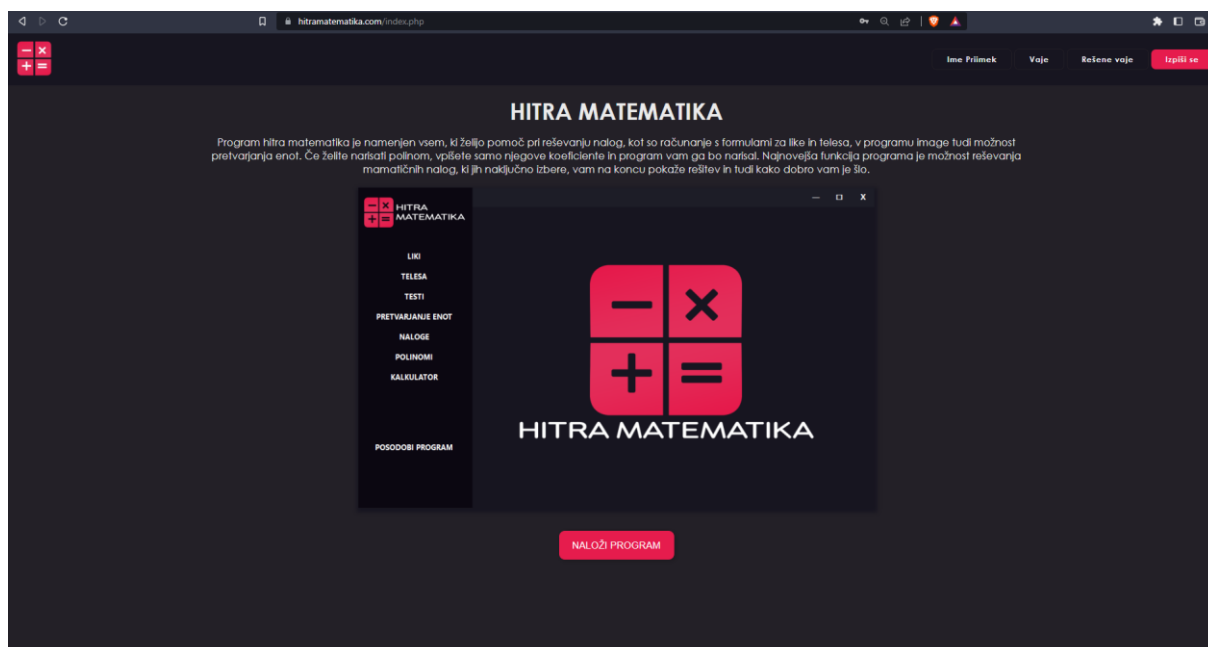


Slika 27: Nameščena datoteka na namizju.

5 Spletna stran

5.1 Predstavitev

Odločila sva se, da bo spletna stran omogočala registracijo in vpis uporabnikov. Ob vpisu lahko uporabnik dostopa do programa, ki si ga lahko naloži ob kliku na gumb. Na spletni strani je možno tudi reševanje nalog, ki se naključno izberejo iz podatkovne baze. Ob uspešni oddaji se na podlagi pravih odgovorov, ki so zapisani v podatkovni bazi, izračuna število pravih odgovorov, poleg tega pa se izpiše, koliko odstotkov je uporabnik dosegel, kako dolgo je reševal celotno vajo in kdaj jo je oddal. Vsak uporabnik ima možnost pregleda rešenih vaj z namenom, da se bo lahko učil iz svojih napak.

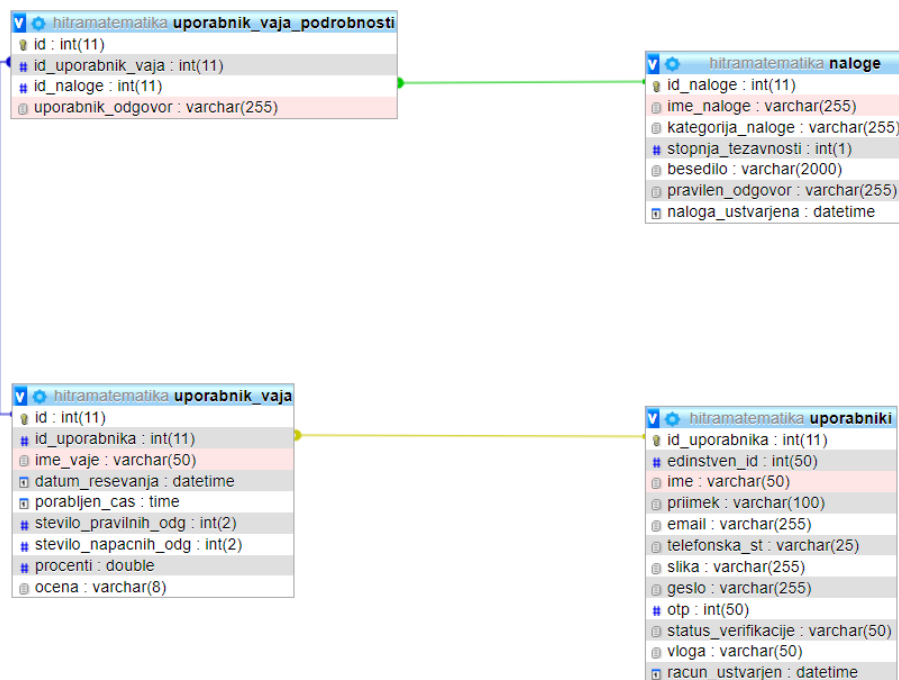


Slika 28: Osnovna stran.

Do spletne strani lahko dostopate preko <https://hitramatematika.com>

5.2 Načrtovanje podatkovne baze

Za načrtovanje in izdelavo podatkovne baze sva uporabila phpMyAdmin, ki omogoča enostavno izdelavo ter upravljanje s podatkovnimi bazami in tabelami v njih. Najprej sva si na list papirja skicirala, kako bo izgledala postavitev, naštela vse attribute v vsaki tabeli ter določila primarne in tuje ključe, da sva lahko ustvarila ustrezno povezavo med tabelami.



Slika 29: ER diagram podatkovne baze.

5.3 Gostovanje

Odločila sva sem da bova za gostovanje spletne strani uporabila GoDaddy, saj je ena največjih in najbolj znanih spletnih gostovanj na svetu. Poleg tega ponuja zelo konkurenčne cene, zanesljivost in enostavno uporabo. GoDaddy ima tudi širok nabor funkcij in orodij, ki jih lahko uporabiva za izboljšanje delovanja najine spletne strani, kot so SSL certifikati, hitrost strani in podpora za različne platforme. Prepričana sva, da bo z uporabo GoDaddyja najina spletna stran zanesljiva in hitra za uporabnike.

5.4 Registracija in prijava

Ob vpisu <https://hitramatematika.com> v brskalnik uporabnik vidi stran za registracijo, kjer mora vpisati ime, priimek, e-naslov, telefonsko številko, geslo ter izbrati profilno sliko. V primeru, da so vsi podatki v pravilni obliki, bo ob kliku na gumb »registrirajte se« uporabnika preusmerilo na stran, kjer bo moral vnesti štirimestno potrditveno kodo, ki jo prejme na e-naslov. Če je koda pravilna, bo uporabnik dobil dostop do spletne strani, v podatkovni bazi pa se bo atribut »status verifikacije« nastavil na »potrjen«.

```

if(in_array($img_extension, $extensions) == true){
    $time = time();
    $newimagename = $time . $img_name;
    if(move_uploaded_file($tmp_name, "../Images/" . $newimagename))
    {
        $random_id = rand(time(), 10000000);
        $otp = mt_rand(1111,9999);

        $sql2 = mysqli_query($conn,"INSERT INTO uporabniki (edinstven_id, ime, priimek, email, telefonska_st, geslo, slika, otp, status_verifikacije, vloga)
        VALUES ('{$random_id}', '{$fname}', '{$lname}', '{$email}', '{$phone}', '{$password}', '{$newimagename}', '{$otp}', '{$verification_status}', '{$role}')");

        if ($sql2) {
            $sql3 = mysqli_query($conn, "SELECT * FROM uporabniki WHERE email = '{$email}'");
            if (mysqli_num_rows($sql3) > 0) {
                $row = mysqli_fetch_assoc($sql3);
                $_SESSION['edinstven_id'] = $row['edinstven_id'];
                $_SESSION['email'] = $row['email'];
                $_SESSION['otp'] = $row['otp'];

                if ($otp) {
                    $receiver = $email;
                    $subject = "From: $fname $lname <{$email}>";
                    $body = "Ime in priimek: ".$fname $lname . " Vaš email ".$email . " Vaša potrditvena koda ".$otp . " Hvala za kreiranje računa na naši spletni strani ";
                    $sender = "From: potrditev@hitramatematika.com";

                    if(mail($receiver, $subject, $body, $sender)) {
                        echo "success";
                    }
                    else{
                        echo "Problem z email naslovom". mysqli_error($conn);
                    }
                }
            }
            else{
                echo "Nekaj je slo narobe!";
            }
        }
        else{
            echo "Prosim izberite profilno sliko - JPG, PNG, JPEG";
        }
    }
}

```

Slika 30: Izsek kode za registracijo.

Po uspešni registraciji in vpisu pravilne kode se lahko uporabnik vpiše na spletno stran ter tako dostopa do njenih funkcij. Trenutno lahko rešuje vaje, ki so sestavljene iz naključnih nalog, jih odda, na koncu pa preveri rezultate.

5.5 Nalaganje programa

Po vpisu uporabnik pridobi dostop do programa, ki ga lahko naloži na osebni računalnik z Windows operacijskim sistemom. Ob kliku na gumb lahko izbere, kam želi naložiti datoteko za nalaganje, in jo kasneje zažene.

5.6 Reševanje vaj

Ob kliku na gumb »vaje«, ki se nahaja v navigacijski vrstici, se na spletni strani odpre stran za reševanje vaj, kjer se iz podatkovne baze izberejo naključne naloge. Po desetih sekundah se sproži štoparica, ki meri čas, ustavi pa se ob uspešni oddaji vaje. Preden lahko uporabnik odda nalogo, jo mora najprej poimenovati, potem pa rešiti vse dane naloge. Odgovor mora biti v pravilnem formatu, sicer se bo štel kot napačen. V prihodnosti bova dodala tudi algoritem za preverjanje pravilnosti odgovora, saj uporabnikov odgovor ne bo vedno enak kot tisti, ki je shranjen v podatkovni bazi.

Matematični program

REŠI VAJO

00:00:05
Shrani vajo kot:

Vsota polinomov - POLINOMI

Izračunajte vsoto $(3x^2 + 2x - 1) + (2x^2 - 4x + 5)$. Odgovor zapišite v obliki: $2x^2 + x + 5$

Vnesi rešitev vaje

Obseg kvadrata - GEOMETRIJA

Izračunajte obseg kvadrata s stranico 5cm. Odgovor zapišite v obliki: x cm

Vnesi rešitev vaje

Množenje - POLINOMI

Pomnožite $(2x - 3)$ s $(4x + 5)$. Odgovor zapišite v obliki: $2x^2 + x + 5$

Vnesi rešitev vaje

g v kg - PRETVARJANJE ENOT

Pretvorite 500g v kg. Odgovor zapišite v obliki: x kg

Vnesi rešitev vaje

Pitagorov izrek - GEOMETRIJA

Izračunajte dolžino hipotenuze pravokotnega trikotnika, kjer sta kateti dolgi 3cm in 4cm. Odgovor zapišite v obliki: x cm

Vnesi rešitev vaje

Oddaj vajo

Poglej rešene vaje

Slika 31: Reševanje vaje.

5.7 Pregled rešenih vaj

Po oddaji lahko uporabnik dostopa do vseh rešenih vaj, jih pregleda in se uči iz svojih napak.

| VAŠE REŠENE VAJE | | | | | |
|------------------|-------------------|-------|------------------------|---------------|------------------|
| IME VAJE | ŠTEVILO PROCENTOV | OCENA | DATUM IN ČAS REŠEVANJA | PORABLJEN ČAS | PODROBNOSTI |
| tes11 | 60 % | 3 | 05. Marec 2023, 15:49 | 00:06:31 | Prikaži odgovore |

Slika 32: Uporabnikove rešene vaje.

Vsaka vaja ima gumb »prikaži odgovore«, ki prikaže podrobnosti rešene vaje. V primeru, da se uporabnikov odgovor in odgovor, ki je zapisan v podatkovni bazi, ujemata, se bo zraven pojavila kljukica, če pa se odgovora ne ujemata, se bo zraven pojavil križec.

test1

Vsota polinomov

Navodilo:
Izračunajte vsoto $(3x^2 + 2x - 1) + (2x^2 - 4x + 5)$. Odgovor zapišite v obliki: $2x^2 + x + 5$

Vaš odgovor: 1 ❌
Pravilen odgovor: $5x^2 - 2x + 4$

Obseg kvadrata

Navodilo:
Izračunajte obseg kvadrata s stranico 5cm. Odgovor zapišite v obliki: x cm

Vaš odgovor: 20 cm ✅
Pravilen odgovor: 20 cm

Množenje

Navodilo:
Pomnožite $(2x - 3)$ s $(4x + 5)$. Odgovor zapišite v obliki: $2x^2 + x + 5$

Vaš odgovor: 2 ❌
Pravilen odgovor: $8x^2 - 2x - 15$

g v kg

Navodilo:
Pretvorite 500g v kg. Odgovor zapišite v obliki: x kg

Vaš odgovor: 0.5 kg ✅
Pravilen odgovor: 0.5 kg

Pitagorov izrek

Navodilo:
Izračunajte dolžino hipotenuze pravokotnega trikotnika, kjer sta kateti dolgi 3cm in 4cm. Odgovor zapišite v obliki: x cm

Vaš odgovor: 5 cm ✅
Pravilen odgovor: 5 cm

Slika 33: Analiza rešene vaje.

5.8 Izpis uporabnika

Vsak uporabnik ima možnost izpisa, ki poskrbi za to, da se trenutna seja, ki se ustvari ob prijavi, uniči. Uporablja se tudi funkcija `mysqli_real_escape_string()`, ta se uporablja v aplikacijah, ki uporabljajo MySQL bazo podatkov in je namenjena preprečevanju vdorov SQL injekcij. V tem primeru se funkcija uporablja za zaščito parametra "logout_id", ki je del sistema za izpis uporabnika iz spletne strani. Če ta parameter ni zaščiten, lahko zlonamerni uporabnik izkoristi ranljivost in izvede škodljive ukaze v bazi podatkov. Z uporabo funkcije `mysqli_real_escape_string()` se prepreči takšne neželene vplive na sistem.

```
<?php

session_start();
include 'db.php';
if (isset($_SESSION['edinstven_id'])) {
    $logout_id = mysqli_real_escape_string($conn, $_GET['logout_id']);
    if (isset($logout_id)) {
        session_unset();
        session_destroy();
        header("location: ../login.php");
    }
    else{
        header("location: ../index.php");
    }
}
else{
    header("location: ../login.php");
}

?>
```

Slika 34: Izpis uporabnika.

6 Ovrednotenje hipotez

V nadaljevanju bova ovrednotila hipoteze, ki sva jih postavila na začetku najinega raziskovanja in so naju vodile pri izdelavi programa. Vsaka od hipotez se nanaša na določeno področje delovanja programa, od potenciala za poučevanje, do zahtevanih knjižnic in stroškov izdelave.

H1: Profesorji bodo program prepoznali kot primeren za poučevanje.

Po pogovoru s profesorico matematike, ki je program preizkusila, lahko rečeva, da je bil program ocenjen kot primeren za poučevanje. Ugotovila sva, da ima program potencial za poučevanje in reševanje nalog, vendar bi bilo potrebnih nekaj popravkov pri razlagi snovi za like, da bi se program popolnoma ujemal z učnimi načrti in standardi za področje matematike.

Na podlagi pozitivnega mnenja profesorice, obstaja potencial, da bodo tudi drugi profesorji prepoznali njegovo uporabnost in ga ocenili kot primerne za poučevanje.

To hipotezo lahko potrdiva.

H2: Program bo omogočal reševanje matematičnih nalog, kot so geometrija, trigonometrija, polinomi ...

To hipotezo lahko potrdiva, saj najin program omogoča vse naštetu. Omogoča računanje z liki in telesi, risanje linearne in kvadratne funkcije, polinomov tretje, četrte in pete stopnje ter logaritemske funkcije. V prihodnosti želiva dodati tudi eksponentno in racionalno funkcijo, za kateri nama je zmanjkalo časa. Pri racionalni funkciji se je pojavil problem pri polih, saj se graf poveže, kjer se ne bi smel.

Tudi to hipotezo lahko potrdiva.

H3: Za pravilno delovanje programa, bodo morale biti vključene različne knjižnice.

Glede na dejstvo, da v .NET Frameworku ni na voljo dovolj knjižnic za risanje grafov, je to hipotezo mogoče potrditi. Za pravilno risanje grafov sva morala uporabiti knjižnico ZedGraph, ki omogoča uporabniku več možnosti upravljanja z grafom. Uporabila sva tudi knjižnico MySql.Data, ki omogoča dostop do podatkovnih zbirk, Newtonsoft.Json, ki pomaga pri komunikaciji s podatkovno bazo, ZXing za QR kode in Otp.NET za pošiljanje avtentikacijske kode uporabniku.

H4: Za izdelavo programa bo zadoščalo znanje, pridobljeno v šoli.

To hipotezo lahko samo delno potrdiva, saj sva za izdelavo API-ja potrebovala znanje, ki ga nisva pridobila v šoli. Tudi pri risanju grafov sva potrebovala znanje o lambda operatorjih, ki ga v šoli še nismo usvojili.

H5: Stroški bodo nastali samo pri gostovanju spletne strani.

Edini stroški, ki so se pojavili pri izdelavi programa in spletne strani, so bili za gostovanje na strani: <https://www.godaddy.com>. Poleg tega sva zraven gostovanja dobila tudi dva profesionalna e-naslova s končnico: »@hitramatematika.com« ter domeno: »hitramatematika.com«.

Hipotezo lahko torej potrdiva.

7 Zaključek

Na podlagi raziskovalne naloge, v kateri sva izdelala program za računanje podatkov o likih in telesih, ugotavljava, da je bil projekt uspešno dokončan. Program omogoča uporabnikom preprosto izračunavanje podatkov o likih in telesih, risanje raznih funkcij ter pretvarjanje enot.

V najin program sva dodatno vključila možnost risanja polinomskih grafov, kar uporabnikom omogoča vizualno predstavitev matematičnih konceptov in idej. Pri izdelavi programa sva se osredotočila tudi na uporabnost in varnost, saj sva implementirala funkcije za prijavo in dvostopenjsko overjanje, ki zagotavljata, da imajo dostop do programa le pristojni uporabniki.

Poleg tega sva ustvarila tudi spletno stran, ki omogoča uporabnikom, da naložijo program in rešujejo naloge za vajo kar preko spleta. Uporabnikom sva tako omogočila, da lahko dostopajo do programa na enostaven način, kar bo zagotovo pripomoglo k večji uporabnosti programa. Na spletni strani lahko uporabniki preizkusijo svoje matematično znanje in si ogledajo svoje dosežke ter napredek. S tem sva želela ustvariti celovit program za reševanje matematičnih problemov.

V prihodnje si želiva nadgraditi program z novimi funkcijami in ga narediti uporabniku še bolj prijaznega.

Spletna stran, ki sva jo ustvarila, bo omogočila dostop do programa še večjemu številu ljudi, saj bo dostopen preko spleta, kar je zelo priročno. Verjameva, da bo najina aplikacija imela pozitiven učinek na izobraževanje in širjenje znanja na področju matematike.

8 Literatura

- Anh, L. T. (3. marec 2021). *YouTube*. Pridobljeno 21. januar 2023 iz C# Windows Form Application Call PHP Web Service:
https://www.youtube.com/watch?v=kUvaE_v30IU
- Brilej, R. (2011). *ALFA. POLINOMI IN RACIONALNE FUNKCIJE*. Ljubljana: Ataja, d.o.o.
- EN, R. C. (26. julij 2021). *YouTube*. Pridobljeno 3. februar 2023 iz Dropdown Menu - Custom ContextMenuStrip (Multilevel) - C# & WinForms:
<https://www.youtube.com/watch?v=2h69Ce4MZiQ>
- Gigi. (19. Oktober 2019). *USING TIME-BASED ONE-TIME PASSWORDS FOR TWO-FACTOR AUTHENTICATION*. Pridobljeno 12. Januar 2023 iz Gigi Labs:
<https://gigi.nullneuron.net/gigilabs/using-time-based-one-time-passwords-for-two-factor-authentication/>
- gwycon. (16. januar 2009). *YouTube*. Pridobljeno 6. februar 2023 iz Polynomial Explorer - C# Application in Visual Studio 2008:
<https://www.youtube.com/watch?v=gkHqBZSfKEA>
- jchampion, k. r. (19. september 2012). *SOURCEFORGE*. Pridobljeno 12. december 2022 iz ZedGraph: <https://sourceforge.net/projects/zedgraph/>
- Kebe Krhin, N. (2019). *Matzapiski za gimnazije : tudi za maturo*. Novo mesto : v samozal.
- Learn, F. (13. april 2017). *YouTube*. Pridobljeno 1. februar 2023 iz C# Tutorial - Metro Login Form with SQL Database Step By Step | FoxLearn:
<https://www.youtube.com/watch?v=ruTifUtQu88>
- Legiša, P. (1990). *Matematika. Polinomi, racionalne funkcije, krivulje drugega reda*. Ljubljana: Državna založba Slovenije.
- Mortensen, P. (3. September 2012). *stack overflow*. Pridobljeno iz Hiding the gray border of the ZedGraph control: <https://stackoverflow.com/questions/12181556/hiding-the-gray-border-of-the-zedgraph-control>
- Srečo, L., & Matija, U. (29. April 2016). *Wikipedijski*. Pridobljeno 1. marec 2023 iz Programski jezik C sharp: https://sl.wikipedia.org/wiki/Programski_jezik_C_sharp

Winney, G. (15. avgust 2019). *grantwinney*. Pridobljeno 23. december 2022 iz How to generate and verify TOTP 2FA codes (in C#): <https://grantwinney.com/a-sample-csharp-app-for-generating-and-verifying-totp-2fa-codes/>

9 Priloge

9.1 Anketni vprašalnik

Razdelek 1 od 3

Hitra matematika

Program hitra matematika bo omogočal računanje podatkov za like, telesa... Možno bo tudi reševanje nalog, testov, pretvarjanje enot in tudi risanje polinomov. Da bi nama pomagali pri izboljšanju programa, prosim rešite spodnjo anketo.

Po razdelku 1 Naprej na naslednji razdelek

Razdelek 2 od 3

Splošno

Opis (neobvezno)

Kako dobro ti gre matematika (1-5) *

☐ 1

☐ 2

☐ 3

☐ 4

☐ 5

Ali potrebuješ program, ki bo imel vse na enem mestu? *

☐ Da

☐ Ne

S čim imaš pri matematiki največ težav (kaj najbolj potrebuješ v programu)? *

Besedilo vprašanja z dolgim odgovorom

Meniš, da bi se s pomočjo tega programa lažje in hitreje učil/a? *

☐ Da

☐ Ne

☐ Drugo ...

Razdelek 3 od 3

Ostalo



Opis (neobvezno)

Po tvojem mnenju misliš, da bi bil ta program koristen? *

Besedilo vprašanja s kratkim odgovorom

Ali bi tak program redno uporabljal/a? *

- ☐ Da
- ☐ Ne
- ☐ Drugo ...

Katerega spola si? *

- ☐ Moški
- ☐ Ženska

Koliko si star/a *

- ☐ do 15 let
- ☐ od 15 do 19 let
- ☐ več kot 19 let

IZJAVA*

Mentor/-ica Boštjan Resinovič v skladu z 20. členom Pravilnika o organizaciji mladinske raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi z naslovom matematični program, katere avtor/-ica je Blaž Hribernik in Urban Levičar:

- besedilo v tiskani in elektronski obliki istovetno,
- pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,
- da je za objavo fotografij v nalogi pridobljeno avtorjevo dovoljenje in je hranjeno v šolskem arhivu,
- da sme Osrednja knjižnica Celje objaviti raziskovalno nalogo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloga nastala v okviru projekta Mladi za Celje,
- da je raziskovalno nalogo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
- da smo seznanjeni z razpisni pogoji projekta Mladi za Celje.

Celje, 4.4.2023



Podpis mentorja

Podpis odgovorne osebe

*

POJASNILO

V skladu z 20. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje je potrebno podpisano izjavo mentorja (-ice) in odgovorne osebe šole vključiti v izvod za knjižnico, dovoljenje za objavo avtorja (-ice) fotografskega gradiva, katerega ni avtor (-ica) raziskovalne naloge, pa hrani šola v svojem arhivu.