

**PANEVROPSKI UNIVERZITET APEIRON, BANJA LUKA
FAKULTET INFORMACIONIH TEHNOLOGIJA**

Redovne studije

Smjer : “Inženjering Informacionih Tehnologija“

Predmet:

Viši programski Jezici i RAD Alati

**Zadaci iz predmeta „Viši programski jezici i RAD
alati“**

(Seminarski rad)

Predmetni nastavnik

Prof. dr Zoran Ž. Avramović dipl.inž.elek.

Student: Pavlović Ivan

Br. Indeksa : 92-20/RITP-S

Banja Luka, Septembar 2023.

SADRŽAJ

UVOD	1
1. Visual Basic.NET	2
1.1. Visual Basic.NET Sintaksa.....	3
1.2. .NET implementacije	3
1.3. Veb okviri (engl. Web Frameworks) u VB	4
1.4. Grafički korisnički interfejs u VB	5
2. Zadaci u Visual Basic.NET.....	5
2.1. Visual Basic.NET zadaci u konzoli (Console Apps)	5
2.1.1. Provjera da li je unešeni broj paran ili neparan u konzoli.....	5
2.1.2. Unos stringa od korisnika koji se ispisuje unazad	6
2.1.3. Pogoditi nasumično generisan broj između 1 i 15	8
2.1.4. Najduža i najkraća rečenica u korisnički unesenom stringu	10
2.1.5. Kalkulator u VB.NET konzoli	12
2.2. Visual Basic.NET zadaci u Windows Formama (Windows Forms Application).....	14
2.2.1. BMI Kalkulator	14
2.2.2. Digitalni Sat	17
2.2.3. Konverter valuta.....	19
2.2.4. Analiza Teksta	22
2.2.5. Fibonačijev niz.....	24
2.2.6. Čitač tekstualnih datoteka	26
2.2.7. Aritmetička sredina	28
2.2.8. Najveći broj od unesenih brojeva	29
2.2.9. Jednostavna Paint alatka	30
2.2.10. Izračun mjesečne rate kredita	32
2.2.11. Bubble Sort.....	33
2.2.12. Faktorijel broja	35
2.2.13. Kalkulator potrošnje putovanja	36
2.2.14. Kalkulator brzine tipkanja.....	37
2.2.15. Generator lozinke	38
2.2.16. Kalkulator životnog vijeka baterije	39
2.2.17. Prikaz specifikacija operativnog sistema	40
2.2.18. Prikaz teksta u različitim bojama	41
2.2.19. Provjera da li je broj prost	42

2.2.20.	Pretvarač mjernih jedinica.....	44
2.2.21.	Rečenica bez razmaka	45
2.2.22.	Generator nasumičnih brojeva od 1 do 50	46
2.2.23.	Kalkulator.....	47
2.2.24.	Kalkulator razlike kvadrata	50
2.2.25.	Najveći broj	52
ZAKLJUČAK.....		53
POPIS SLIKA.....		54
CITATNI IZVORI		55

UVOD

U savremenom svijetu softvera, programiranje je postalo neophodan alat za razvoj različitih aplikacija koje olakšavaju život, poboljšavaju produktivnost i omogućavaju kreativnost. U okviru ovog seminarskog rada, prikazaću zadatke u Visual Basic.NET (VB.NET) programskom jeziku, koji se ističe kao moćan i fleksibilan alat za razvoj aplikacija na platformi Windows.

Objektno orijentisano programiranje (OOP) postaje sve značajnije u razvoju softvera zbog svoje sposobnosti organizovanja koda u objekte sa definisanim svojstvima i metodama. Visual Basic NET, kao jezik baziran na OOP principima, omogućava programerima kreiranje modularnih, skalabilnih i održivih aplikacija.

U cilju brzog razvoja aplikacija, koristićemo alate poznate kao RAD (engl. Rapid Application Development) alate. Oni će nam omogućiti kreiranje aplikacija sa minimalnim kodiranjem, oslanjajući se na vizuelno okruženje i gotove komponente. U ovom radu, posebno ćemo se fokusirati na razlike između dve glavne vrste aplikacija koje se mogu razviti u VB.NET (konzolnih aplikacija) i VB Forme (aplikacija sa grafičkim korisničkim interfejsom).

1. Visual Basic.NET

„Visual Basic (VB), prvobitno nazvan Visual Basic.NET, je višeparadigmatski objektno orijentisan programski jezik, implementiran na .NET, Mono i .NET framework (engl. Framework).

Microsoft je pokrenuo VB.NET 2002. godine kao naslijednika svog originalnog jezika Visual Basic, čija je posljednja verzija bila Visual Basic 6.0. Iako je „.NET“ dio imena izbačen 2005. godine ali se idalje koristi kako bi se napravila razlika između njih i klasičnog Visual Basic-a. Zajedno sa C# i F#, to je jedan od tri glavna jezika koji ciljaju na .NET ekosistem. Microsoft je ažurirao svoju VB jezičku strategiju 6. februara 2023. godine, navodeći da je VB sada stabilan jezik i da će se Microsoft nastaviti da ga održava.

Microsoftovo razvojno oružanje (IDE) za razvoj u Visual Basic-u je Visual Studio. Većina izdanja Visual Studio-a je komercijalna, a jedini izuzetci su Visual Studio Express i Visual Studio Community, koji su besplatni. Pored toga, .NET Framework SDK uključuje besplatni kompajler komandne linije pod nazivom „vbc.exe“.



Slika 1 - Visual Basic.NET logo i logotip

Visual Basic se često koristi u kombinaciji sa Windows formama bibliotekom za pravljenje desktop aplikacija za Windows. Programiranje Windows formi sa Visual Basic-om uključuje prevlačenje i ispuštanje kontrola na ekranu pomoću GUI dizajnera i pisanjem odgovarajućeg koda za svaku kontrolu“ [1].

1.1. Visual Basic.NET Sintaksa

„Visual Basic koristi iskaze (engl. statements) da specificira radnje. Najčešći iskaz je ekspresioni iskaz, koji se sastoji od iskaza koji se vrijednuje u jednom redu. Kao dio te evaulacije, funkcije ili podprogrami mogu biti pozvani i promjenljivima (engl. variables) se mogu dodijeliti nove vrijednosti.

Da bi modifikovali normalnu sekvencijalno izvršavanje naredbi, Visual Basic obezbijeduje nekoliko naredbi toka kontrole identifikovanih rezervisanim ključnim riječima. Strukturirano programiranje je podržano sa nekoliko konstrukcija, uključujući dve konstrukcije uslovnog izvršenja (If... Then... Else... End If... i Select Case... Case... End Select...) i tri iterativne konstrukcije izvršenja tzv. petlje (Do... Loop..., For... To, i For Each...).

Ne postoji jedinstven način definisanja blokova iskaza. Umjesto toga, određene ključne riječi, kao što su „If Then“ ili „Sub“ se tumače kao početak podblokova koda i imaju odgovarajuće ključne riječi za završetak, kao što su „End If“ i „End Sub“.

Iskazi (engl. Statements) se završavaju dvotačkom ili krajem reda. Iskazi sa više redova u Visual Basic-u su omogućene sa „_“ na kraju svakog takvog reda. Potreba za nastavkom donje crte je u velikoj mjeri uklonjena u verziji 10 i kasnijim verzijama.

Znak jednakosti „=“ se koristi za dodijeljivanje vrijednosti i za poređenje. Okrugle zagrade se koriste sa nizovima, kako za njihovo deklarisanje, tako i za dobijanje vrijednosti na datom indeksu u jednom od njih. Visual Basic koristi zagrade za definisanje parametara podprograma ili funkcija. Jedan navodnik ili ključna riječ REM, stavljen na početak reda ili poslije bilo kog broja znakova za razmak ili uvlaka na početku reda, ili poslije drugog koda na redu, označava da je (ostatak) reda komentar“ [1].

1.2. .NET implementacije

„U sadašnjem vremenu postoje četiri primarne .NET implementacije koje se aktivno održavaju i razvijaju:

.NET Framework – originalna .NET implementacija koja postoji još od 2002. godine. Iako još nije ukinuta, Microsoft ne planira da objavi svoju slijedeću veliku verziju 5.

Mono – Višeplatformska implementacija .NET framework-a od strane Xamarin, predstavljena 2004. godine. Mono je otvorenog koda i besplatan je i trenutno ga održava Xamarin, podkompanija Microsoft.

Universal Windows Platform (UWP) – Implementacija .NET-a koja se koristi za pravljenje UWP aplikacija. Dizajniran je da objedini razvoj za različite ciljne uređaje, uključujući računare, mobilne telefone, tablete i Xbox konzole.

.NET – reimplementacija .NET Framework-a na više platformi, predstavljena 2016te godine i prvobitno nazvana .NET Core.

.NET standard je skup uobičajenih API-ja koji su implementirani u biblioteci osnovnih klasa bilo koje .NET implementacije. Biblioteka klasa svake implementacije mora implementirati .NET standard, ali također može implementirati dodatne API-je. Tradicionalno .NET aplikacije su ciljale određenu verziju .NET implementacije, npr. .NET Framework 4.6.

Počevši od .NET standarda, aplikacija može ciljati verziju .NET standarda, a zatim je može koristiti (bez ponovnog kompajliranja) bilo koja implementacija koja podržava taj nivo standarda. Ovo omogućava prenosivost kroz različite .NET implementacije“ [1].

1.3. Veb okviri (engl. Web Frameworks) u VB

„**ASP.NET** – Okvir veb aplikacija na strani servera dizajniran za veb razvoj i proizvodnju dinamičkih veb stranica. To je nasljednik Microsoft-ove tehnologije „Active Server Pages (ASP)“

izgrađene na „Common Language Runtime (CLR)“. APS.NET obezbijuje zasebne obrazce za razvoj veb aplikacija ASP.NET MVC, ASP.NET Web API i ASP.NET Web Pages.

APS.NET Core – naslijednik i ponovna reimplementacija ASP.NET-a kao modularnog veb okvira, zajedno sa drugim okvirima kao što je „Entity Framework“. Ovaj veb okvir koristi novu platformu .NET kompajlera otvorenog koda i višestruko je platformski.

Blazor – je besplatan veb okvir otvorenog koda koji omogućava programerima da kreiraju veb aplikacije koristeći C# i HTML. Blazor serverske aplikacije se hostuju na APS.NET Core serveru u ASP.NET Razor formatu, dok su Blazor WebAssembly apps aplikacije na jednoj stranici koje se preuzimaju u veb pregledač klijenta prije pokretanja“ [2].

1.4. Grafički korisnički interfejs u VB

„Windows Forms (WinForms) – Originalna Microsoft implementacija koja radi na Windows operativnim sistemima i omogućava pristup zajedničkim kontrolama Windows korisničkog interfejsa ubacivanjem Windows API-ja u upravljani kod.

Windows Presentation Foundation (WPF) – grafički podsistem za prikazivanje korisničkih interfejsa u Windows aplikacijama od strane Microsofta. Zasnovan je na DirectX i koristi XAML, jezik zasnovan na XML-u, za definisanje i povezivanje različitih elemenata interfejsa. WPF aplikacije se mogu primjeniti kao samostalni desktop program ili hostovati kao ugrađeni objekat na veb sajtu“ [1].

2. Zadaci u Visual Basic.NET

2.1. Visual Basic.NET zadaci u konzoli (Console Apps)

2.1.1. Provjera da li je unešeni broj paran ili neparan u konzoli


```

Module MainModule
Sub Main()
    Console.WriteLine("Unesite broj:")
    Dim uneseniBroj As Integer = Integer.Parse(Console.ReadLine())

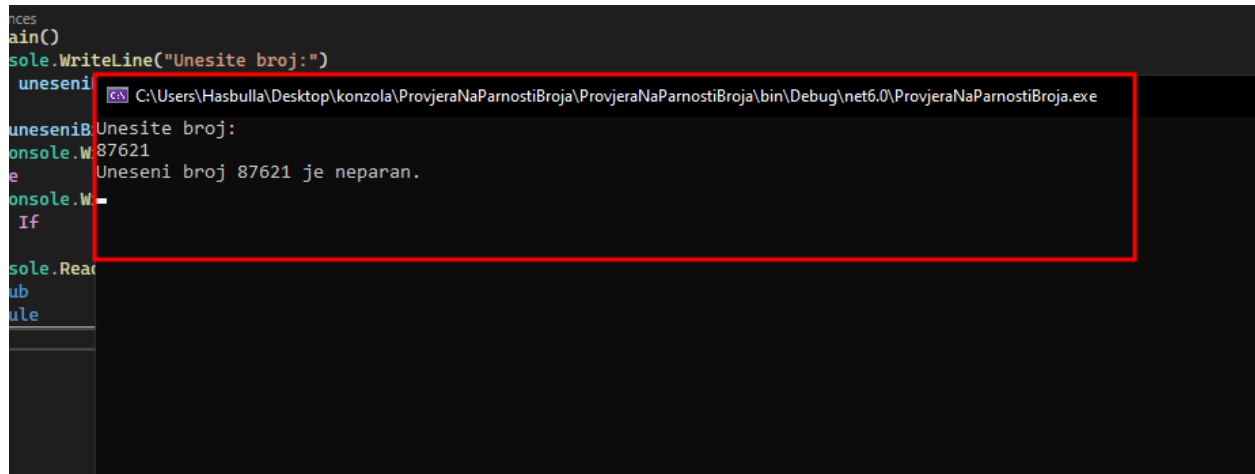
    If uneseniBroj Mod 2 = 0 Then
        Console.WriteLine("Uneseni broj " & uneseniBroj & " je paran.")
    Else
        Console.WriteLine("Uneseni broj " & uneseniBroj & " je neparan.")
    End If

    Console.ReadLine()
End Sub
End Module

```

U ovom programu od korisnika se traži da unese željeni broj koji će se provjeriti da li je paran ili neparan. Funkcija **Integer.Parse()** je metoda u Visual Basic.NET koji je koristi za pretvaranje teksta (stringa) u cijeli broj (integer, u ovom programskom jeziku int32).

Uneseni broj se sprema u varijablu **uneseniBroj**, zatim pomoću operatora **Mod** provjeravamo ostatak pri dijeljenju broja sa dva. U slučaju da je ostatak 0, broj je paran, pa ispisujemo u konzoli da je broj paran, a u suprotnom da je broj neparan. Korištena je **Console.ReadLine()** metoda koja čeka korisnički unos koji će zatvoriti prozor.



```

C:\Users\Hasbulla\Desktop\konzola\ProvjeraNaParnostiBroja\ProvjeraNaParnostiBroja\bin\Debug\net6.0\ProvjeraNaParnostiBroja.exe
Unesite broj:
87621
Uneseni broj 87621 je neparan.

```

Slika 2 - Provjera parnosti i neparnosti broja u VB konzoli

2.1.2. Unos stringa od korisnika koji se ispisuje unazad

```

Module MainModule
Sub Main()
    Console.WriteLine("Unesite neki tekst:")
    Dim uneseniTekst As String = Console.ReadLine()

    Dim obrnutiTekst As String = StrReverse(uneseniTekst)

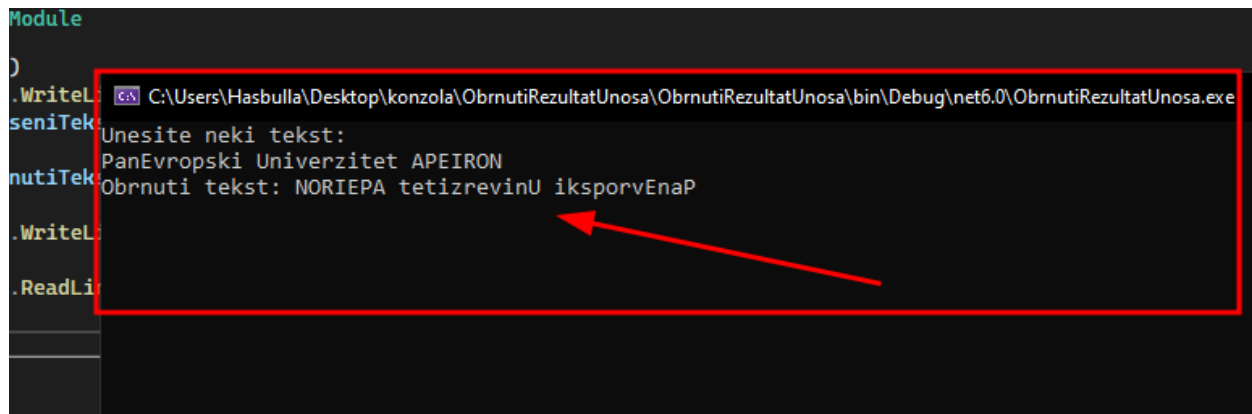
    Console.WriteLine("Obrnuti tekst: " & obrnutiTekst)

    Console.ReadLine()
End Sub
End Module

```

U ovom programu od korisnika se traži da unese neki tekst sa tastature pomoću **Console.ReadLine()** metode. Taj uneseni tekst se čuva u varijabli **uneseniTekst**. Korištenjem ugrađene VB.NET funkcije **StrReverse(uneseniTekst)** koja automatski obrće redoslijed svih znakova u unesenom tekstu i rezultat ove funkcije čuvamo u varijabli **obrnutiTekst**.

Nakon toga, program će ispisati obrnuti tekst korisniku pomoću **Console.WriteLine(„Obrnuti tekst: „ & obrnutiTekst)** i nakon toga aplikacija će čekati da korisnik pritisne Enter prije nego što se program završi pomoću **Console.ReadLine()** metoda koja omogućava korisniku da pročita rezultat prije nego što se aplikacija zatvori.



```

Module
)
.WriteLine: C:\Users\Hasbulla\Desktop\konzola\ObrnutiRezultatUnosa\ObrnutiRezultatUnosa\bin\Debug\net6.0\ObrnutiRezultatUnosa.exe
seniTekst: Unesite neki tekst:
nutiTekst: PanEvropski Univerzitet APEIRON
.WriteLine: Obrnuti tekst: NORIEPA tetizrevinU iksporvEnaP
.ReadLi

```

Slika 3 - Obrtanje unešenog stringa u konzoli

2.1.3. Pogoditi nasumično generisan broj između 1 i 15

```
Module MainModule
Sub Main()
    Dim generatorBroja As New Random()
    Dim broj As Integer = generatorBroja.Next(1, 16)
    Dim pogodjenBroj As Integer

    Console.WriteLine("Pogodite broj izmedju 1 i 15:")

    Do
        Integer.TryParse(Console.ReadLine(), pogodjenBroj)

        If pogodjenBroj = broj Then
            Console.WriteLine("Tacno!")
            Exit Do
        Else
            Console.WriteLine("Netacno, pokusajte ponovo!")
        End If
    Loop While True

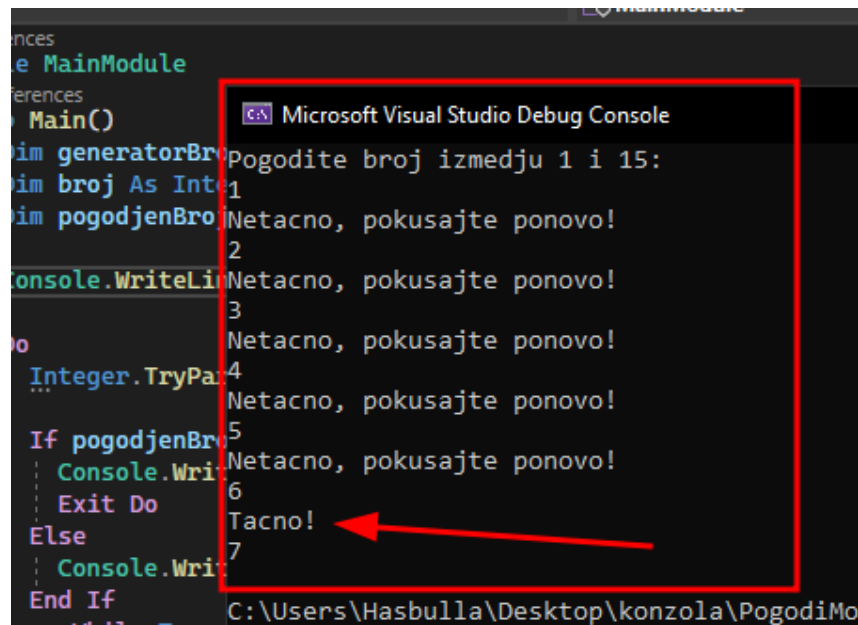
    Console.ReadLine()
End Sub
End Module
```

U ovom programu korisnik mora pogoditi nasumično generisan broj kako bi prekinuo petlju i završio izvršavanje programa. U programu se koristi **Random()** klasa kako bi se stvorio generator nasumičnih brojeva, koji će se koristiti za generisanje nasumičnog broja između 1 i 15 i taj nasumični broj se čuva u varijabli **broj**.

Program će ispisati „Pogodite broj izmedju 1 i 15: „ i zatim čeka korisnikom unos sa tastature. Korisnik unosi broj preko **Console.ReadLine()**, a zatim se pokušava pretvoriti taj unos u cijeli broj koristeći **Integer.TryParse()**, a ukoliko korisnik unese nešto što nije broj, funkcija **TryParse()** će postaviti **pogodjenBroj** na 0.

Nakon korisničkog unosa, program provjerava da li je **pogodjenBroj** jednak generisanom broju koji je u varijabli **broj**. Ako je korisnik pogodio tačan broj, program će ispisati „Tacno!“ i izaći iz **Do** petlje pomoću **Exit Do**. Ako **pogodjenBroj** nije jednak generisanom broju, program će ispisati poruku „Netacno, pokusajte ponovo!“ i korisniku će ponovo omogućiti unos novog broja.

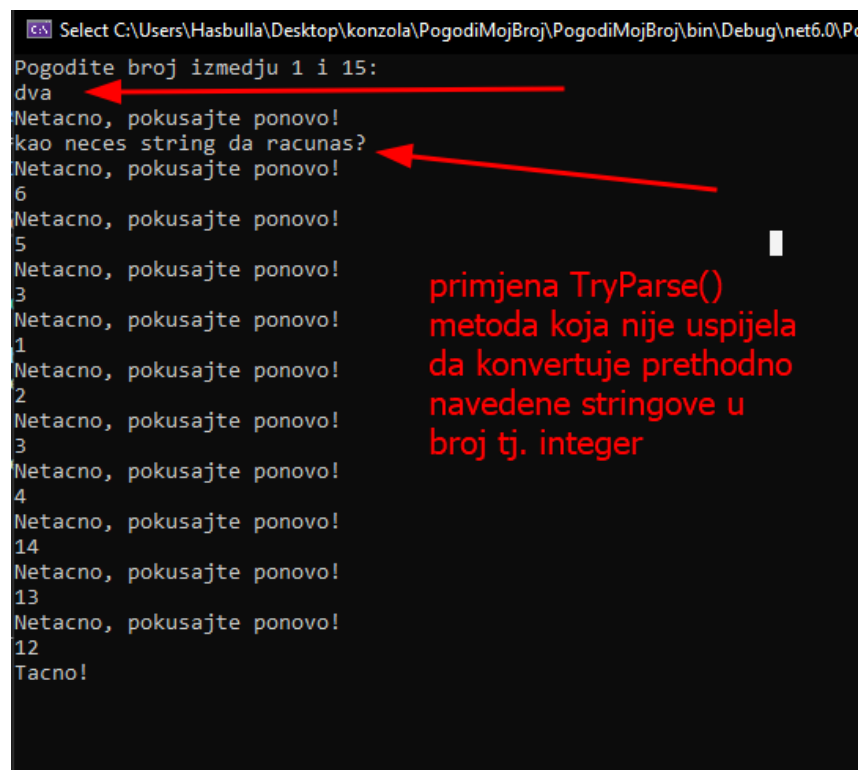
Petlja **Do** će se izvršavati sve dok korisnik ne pogodi tačan broj, radi uslova **While True**, što znači da je petlja beskonačna.



The screenshot shows a C# program in the background and its output in the Microsoft Visual Studio Debug Console. The program is a simple number-guessing game. The console output shows the user entering numbers 1 through 7, all of which are incorrect, followed by the word 'Tacno!' (Correct!). A red box highlights the console output, and a red arrow points to the 'Tacno!' message.

```
Microsoft Visual Studio Debug Console
Pogodite broj izmedju 1 i 15:
1
Netacno, pokusajte ponovo!
2
Netacno, pokusajte ponovo!
3
Netacno, pokusajte ponovo!
4
Netacno, pokusajte ponovo!
5
Netacno, pokusajte ponovo!
6
Netacno, pokusajte ponovo!
7
Tacno!
```

Slika 4 - Rezultat zadatka Br.3



The screenshot shows the execution of a C# program in a command prompt. The user enters the word 'dva' (two) as input. The program prompts the user to guess a number between 1 and 15. The output shows the program repeatedly asking for a guess and responding with 'Netacno, pokusajte ponovo!' (Incorrect, please try again!) until the user enters a correct number. A red arrow points to the input 'dva', and another red arrow points to the program's response. A red text box on the right explains that the `TryParse()` method was used to convert the string input to an integer.

```
Select C:\Users\Hasbulla\Desktop\konzola\PogodiMojBroj\PogodiMojBroj\bin\Debug\net6.0\Po
Pogodite broj izmedju 1 i 15:
dva
Netacno, pokusajte ponovo!
kao neces string da racunas?
Netacno, pokusajte ponovo!
6
Netacno, pokusajte ponovo!
5
Netacno, pokusajte ponovo!
3
Netacno, pokusajte ponovo!
1
Netacno, pokusajte ponovo!
2
Netacno, pokusajte ponovo!
3
Netacno, pokusajte ponovo!
4
Netacno, pokusajte ponovo!
14
Netacno, pokusajte ponovo!
13
Netacno, pokusajte ponovo!
12
Tacno!
```

primjena TryParse()
metoda koja nije uspjela
da konvertuje prethodno
navedene stringove u
broj tj. integer

Slika 5 - Rezultat zadatka Br.3 sa Stringom

2.1.4. Najduža i najkraća rečenica u korisnički unesenom stringu

```
Module MainModule
Sub Main()
    Dim najkracaRecenica As String = Nothing
    Dim najduzaRecenica As String = Nothing

    Console.WriteLine("Unesite niz rijeci (unos završava kada se nista ne unese):")

    While True
        Dim unos As String = Console.ReadLine()

        If String.IsNullOrEmpty(unos) Then
            Exit While
        End If

        If najkracaRecenica Is Nothing OrElse unos.Length < najkracaRecenica.Length
Then
            najkracaRecenica = unos
        End If

        If najduzaRecenica Is Nothing OrElse unos.Length > najduzaRecenica.Length Then
            najduzaRecenica = unos
        End If
    End While

    If najkracaRecenica IsNot Nothing Then
        Console.WriteLine("Najkraca recenica: " & najkracaRecenica)
    End If

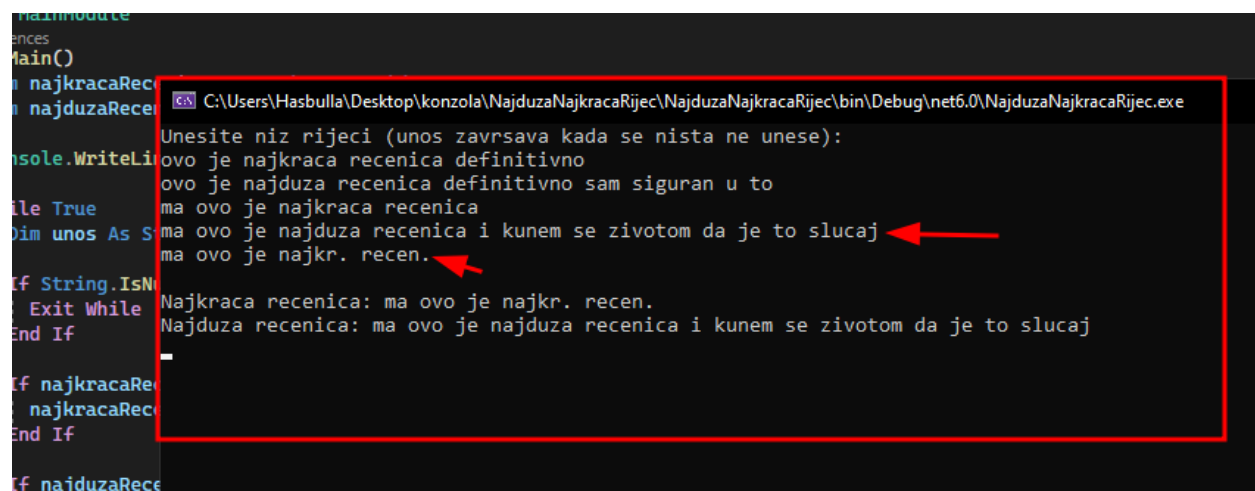
    If najduzaRecenica IsNot Nothing Then
        Console.WriteLine("Najduza recenica: " & najduzaRecenica)
    End If

    Console.ReadLine()
End Sub
End Module
```

U ovom programu definišemo dvije promijenljive **najkracaRecenica** i **najduzaRecenica**, koje su postavljene vrijednosti **Nothing** (null) što znači da trenutno nemaju pronađene najkraće i najduže rečenice. Ispisuje se poruka „Unesite niz rijeci (unos završava kada se nista ne unese): „, kako bi korisnika zamolili da unese sadržaj.

Ulazimo u beskonačnu petlju sa **While True**, što znači da će se unos rečenica odvijati sve dok korisnik ne unese tekst i onda će se petlja prekinuti pomoću **Exit While** naredbe. Korisnik unosi rečenicu pomoću **Console.ReadLine()** i taj unos čuva u promijenljivoj **unos**.

Zatim se vrši provjera da li je korisnik unio tekst i ako jeste izlazimo iz petlje pomoću **Exit While**. Ako trenutno nemamo pronađenu najkraću rečenicu **najduzaRecenica Is Nothing** ili ako je unešena rečenica kraća od trenutno najkraće rečenice, postavljamo **najkracaRecenica** na unijetu rečenicu i isti uslov se vrši na najdužu rečenicu. Nakon što korisnik završi sa unosom, program ispisuje najkraću rečenicu pomoću **Console.WriteLine(„Najkraca recenica: „ & najkracaRecenica)** i najdužu rečenicu pomoću **Console.WriteLine(„Najduza recenica: „ & najduzaRecenica)**.



```

MainModule
ences
Main()
n najkracaRec
n najduzaRec
C:\Users\Hasbulla\Desktop\konzola\NajduzaNajkracaRijec\NajduzaNajkracaRijec\bin\Debug\net6.0\NajduzaNajkracaRijec.exe
Unesite niz rijeci (unos završava kada se nista ne unese):
ovo je najkraca recenica definitivno
ovo je najduza recenica definitivno sam siguran u to
ma ovo je najkraca recenica
ma ovo je najduza recenica i kunem se zivotom da je to slucaj
ma ovo je najkr. recen.
Najkraca recenica: ma ovo je najkr. recen.
Najduza recenica: ma ovo je najduza recenica i kunem se zivotom da je to slucaj
=
If najkracaRec
n najkracaRec
End If
If najduzaRec
```

Slika 6 - Rezultat zadatka Br.4

2.1.5. Kalkulator u VB.NET konzoli

```
Module Module1
Sub Main()
    Console.WriteLine("Dobro dosli u kalkulator!")

    While True
        Console.WriteLine("Unesite prvi broj:")
        Dim broj1 As Double = Double.Parse(Console.ReadLine())

        Console.WriteLine("Unesite operator (+, -, *, /):")
        Dim op As Char = Console.ReadLine()(0)

        Console.WriteLine("Unesite drugi broj:")
        Dim broj2 As Double = Double.Parse(Console.ReadLine())

        Dim rezultat As Double = 0

        Select Case op
            Case "+"
                rezultat = broj1 + broj2
            Case "-"
                rezultat = broj1 - broj2
            Case "*"
                rezultat = broj1 * broj2
            Case "/"
                If broj1 <> 0 Then
                    rezultat = broj1 / broj2
                Else
                    Console.WriteLine("Greska: Ne mozete dijeliti s nulom!")
                    Continue While
                End If
            Case Else
                Console.WriteLine("Greska: Nepoznat operator!")
                Continue While
        End Select

        Console.WriteLine("Rezultat: " & rezultat)

        Console.WriteLine("Zelite li nastaviti? (da/ne)")
        Dim izbor As String = Console.ReadLine().ToLower()

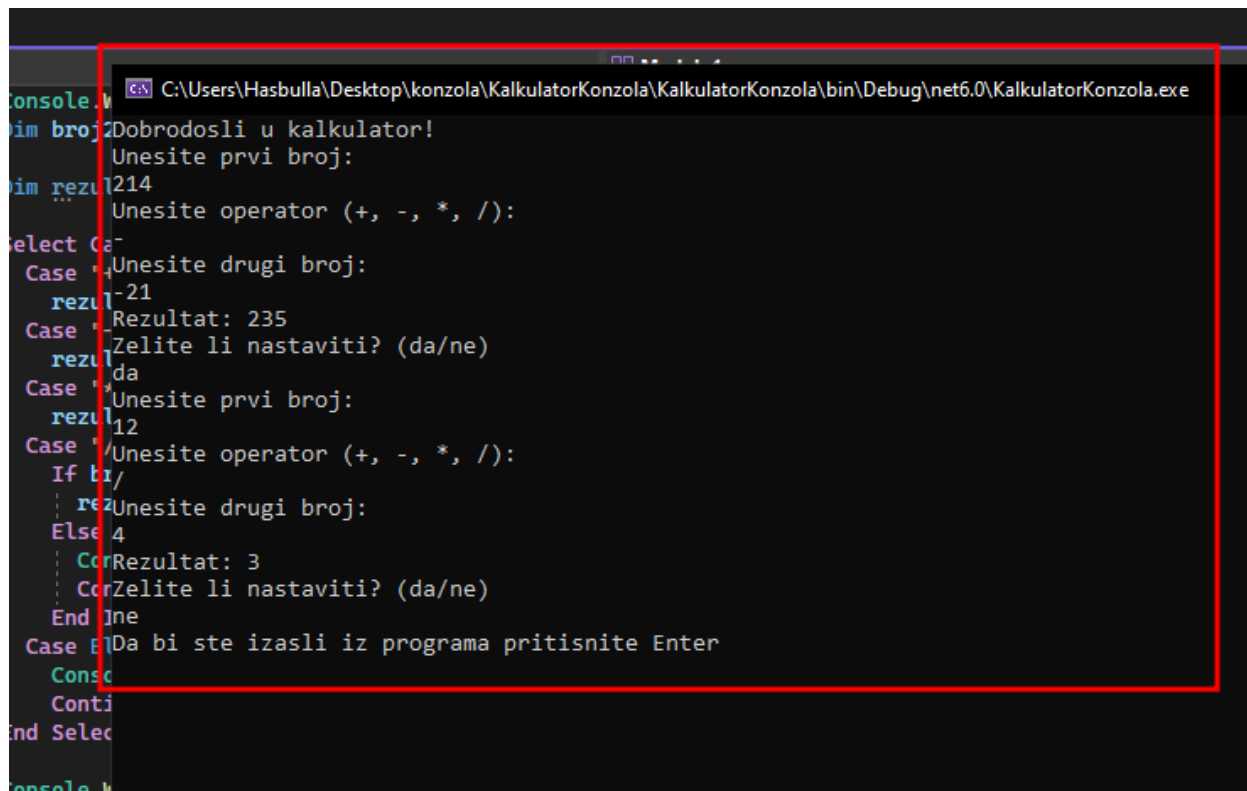
        If izbor <> "da" Then
            Exit While
        End If
    End While

    Console.WriteLine("Da bi ste izašli iz programa pritisnite Enter")
    Console.ReadLine()
End Sub
End Module
```

Program započinje ispisivanjem poruke „Dobro dosli u kalkulator!“ koja pozdravlja korisnika, nakon toga, program ulazi u beskonačnu petlju (**While True**) koja omogućava korisniku da nastavi

s računanjem dok god želi. Unutar petlje, od korisnika se traži da unese prvi broj (**broj1**) koji želi koristiti u računanju. Nakon unosa prvog broja, od korisnika se traži unos aritmetičkog operatora koji određuje operaciju koju želi izvršiti.

Nakon unosa operatora, od korisnika se također traži da unese drugi broj (**broj2**) i rezultat se čuva u varijabli **rezultat** i inicijalizira se na 0. Program koristi **Select Case** uslov za određivanje unesenog operatora i izvođenje određene aritmetičke operacije. Nakon toga, korisniku se postavlja pitanje želi li nastaviti s korištenjem kalkulatora. Ako unos nije „da“, program izlazi iz petlje i nastavlja prema kraju programa.

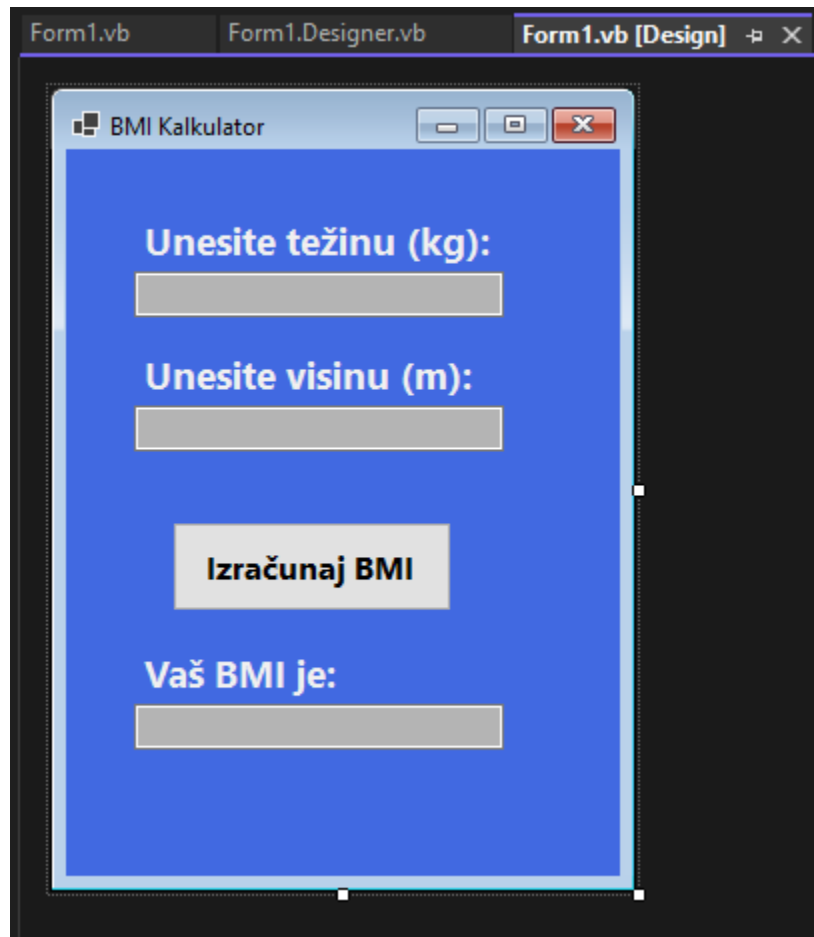


```
Console.WriteLine("Dobro dosli u kalkulator!");
Dim broj1 As Integer
Console.WriteLine("Unesite prvi broj:");
Dim broj2 As Integer
Console.WriteLine("Unesite drugi broj:");
Dim rezultat As Integer = 0
Console.WriteLine("Unesite operator (+, -, *, /):");
Select Case Console.ReadLine()
Case "+"
    rezultat = broj1 + broj2
Case "-"
    rezultat = broj1 - broj2
Case "*"
    rezultat = broj1 * broj2
Case "/"
    rezultat = broj1 / broj2
Case Else
    Console.WriteLine("Neispravan operator")
End Select
Console.WriteLine("Rezultat: {0}", rezultat);
Dim odgovor As String
Console.WriteLine("Zelite li nastaviti? (da/ne)");
If Console.ReadLine().ToLower() = "da" Then
    Console.WriteLine("Unesite prvi broj:");
    Dim broj1 As Integer
    Console.WriteLine("Unesite drugi broj:");
    Dim broj2 As Integer
    Console.WriteLine("Unesite operator (+, -, *, /):");
    If Console.ReadLine().ToLower() = "/" Then
        Console.WriteLine("Ne moze biti deljenja sa nulom");
    Else
        Dim rezultat As Integer = 0
        Console.WriteLine("Rezultat: {0}", rezultat);
        Console.WriteLine("Zelite li nastaviti? (da/ne)");
        If Console.ReadLine().ToLower() = "da" Then
            Console.WriteLine("Da bi ste izašli iz programa pritisnite Enter");
            Console.ReadLine();
        End If
    End If
Else
    Console.WriteLine("Program završava...");
End If
End Select
```

Slika 7 - Rezultat zadatka Br.5

2.2. Visual Basic.NET zadaci u Windows Formama (Windows Forms Application)

2.2.1. BMI Kalkulator



Slika 8 - Prikaz forme u VB Studio Zadatak Br.1

Na početku koda deklariramo dve privatne promijenljive **tezina** i **visina** tipa **Double**. Ove varijable će se koristiti za čuvanje unosa težine i visine korisnika. U događaju **ButtonIzracunaj_Click** koji se aktivira pritiskom na dugme „Izračunaj BMI“, prvo provjerava da li su unijeti podaci za težinu i visinu validni brojevi. To postizemo pomoću metode **Double.TryParse()**, koja pokušava pretvoriti tekst unešen u TextBox „TextBoxTezina“ i „TextBoxVisina“ u numeričke vrijednosti težine i visine. U slučaju da unos nije validan, korisnik

će dobiti grešku na ekranu. Ako korisnik unese validne brojeve, metoda **IzracunajBMI** se poziva sa argumentima tezina i visina kako bi se izračunala vrijednost BMI i ta dobijena vrijednost se kasnije čuva u varijabli **bmi**. Konačan rezultat BMI-a prikazuje se u **TextBoxBMI** i rezultat je zaokružen da dva decimale koristeći metodu **ToString(„0.00“)**. U funkciji **IzracunajBMI**, jednostavno se izračunava BMI formula: $BMI = \text{tezina} / (\text{visina} * \text{visina})$.

```
Public Class Form1
    ' Deklarisemo promijenljive za tezinu i visinu
    Private tezina As Double
    Private visina As Double

    Private Sub ButtonIzracunaj_Click(sender As Object, e As EventArgs) Handles
ButtonIzracunaj.Click
        ' Provjera da li su uneseni brojevi validni
        If Not Double.TryParse(TextBoxTezina.Text, tezina) OrElse Not
Double.TryParse(TextBoxVisina.Text, visina) Then
            MessageBox.Show("Unesite validne brojeve za težinu i visinu!", "Greška",
MessageBoxButtons.OK, MessageBoxIcon.Error)
            Return
        End If
        Dim bmi As Double = IzracunajBMI(tezina, visina)

        TextBoxBMI.Text = bmi.ToString("0.00")
    End Sub

    Private Function IzracunajBMI(tezina As Double, visina As Double) As Double
        Return tezina / (visina * visina)
    End Function

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load

    End Sub
End Class
```

BMI Kalkulator

Unesite težinu (kg):

126

Unesite visinu (m):

1.75

Izračunaj BMI

Vaš BMI je:

41.14

Slika 9 - Rezultat BMI kalkulatora

BMI Kalkulator

Unesite težinu (kg):

126

Unesite visinu (m):

metar

Izračunaj BMI

Vaš BMI je:

41.14

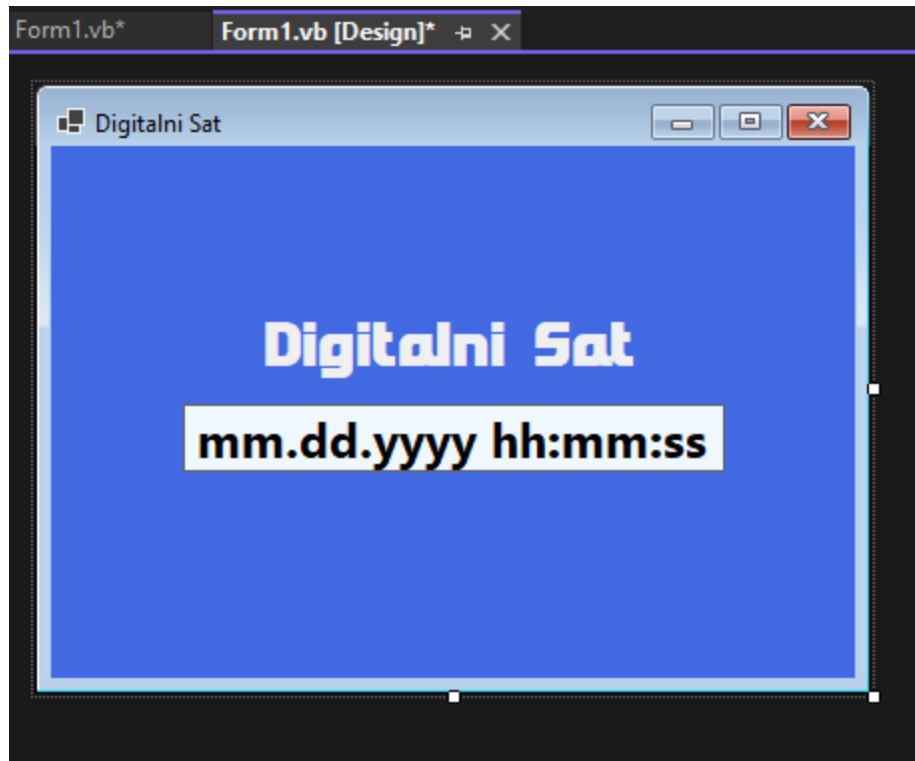
Greška

Unesite validne brojeve za težinu i visinu!

OK

Slika 10 - Korekcija greške u BMI kalkulatoru

2.2.2. Digitalni Sat



Slika 11 - Prikaz forme u VB Studio Zadatak Br.2

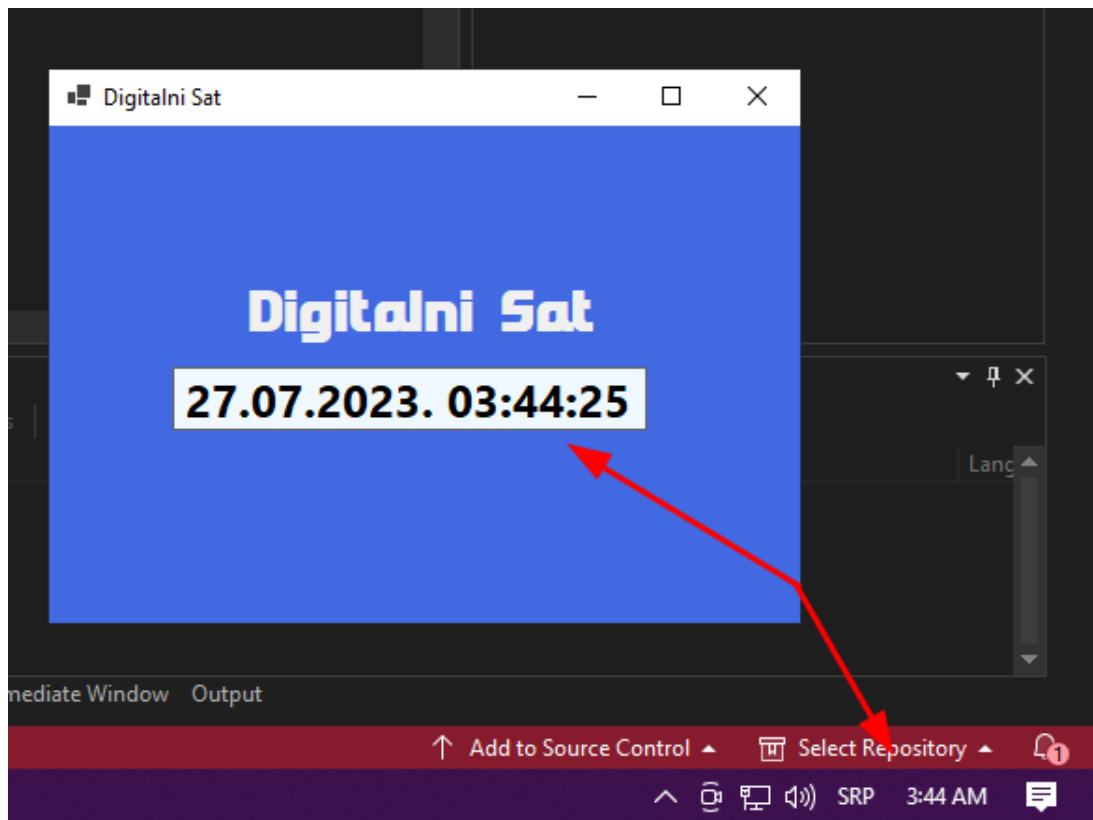
Za ovaj program potrebno je ubaciti **Label** elements koji će prikazivati trenutno vrijeme, format prikaza se može podesiti, ali ja sam odlučio se za format „mm.dd.yyyy hh:mm:ss“. Za prikazivanje vremena potreban nam je element **Timer** kojoj moramo interval pomjeriti sa 100 na 1000 (1 sekunda) jer ćemo ažurirati vrijeme svake sekunde. Zatim, nazivamo **Timer** element „TimerClock“ i postavljamo svojstvo Enabled na „True“.

```
Imports System.Globalization
```

```
Public Class Form1
```

```
    Private Sub timerClock_Tick(sender As Object, e As EventArgs) Handles  
timerClock.Tick  
        ' Ažuriranje trenutnog vremena i datuma u Label kontroli  
        Label1.Text = DateTime.Now.ToString("dd.MM.yyyy. HH:mm:ss",  
CultureInfo.InvariantCulture)  
    End Sub
```

```
End Class
```



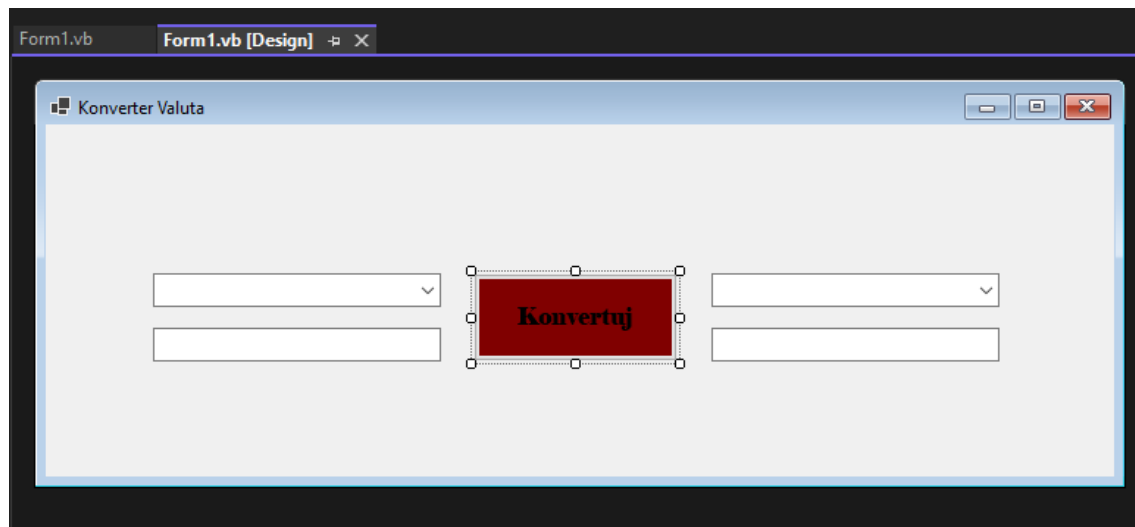
Slika 12 - Rezultat Digitalnog Sata

Private Sub timerClock_Tick(sender As Object, e As EventArgs) Handles timerClock.Tick, ovo je događaj „Tick“ koji se aktivira svaki put kada **Timer** element (timerClock – ime) otkuca. U našem slučaju, postavili smo Timer kontrolu da otkucava svaki sekund, što znači da će se ovaj događaj ažurirati svake sekunde.

Koristimo **DateTime.Now** da bismo dobili trenutni datum i vrijeme sa računara. Metoda **ToString** pretvara DateTime objekat u formatiran niz koji se može prikazati u Label elementu.

CultureInfo.InvariantCulture, je opcionalni argument koji određuje kulturne postavke koje će se koristiti prilikom formatiranja datuma i vremena. Implementovanjem **InvariantCulture**, osiguravamo da formatiranje bude nezavisno od regionalnih podešavanja na računaru tj. bez obzira na to koja je regionalna postavka na računaru, formatiranje će uvijek biti isto.

2.2.3. Konverter valuta



Slika 13 - Prikaz forme u VB Studio Zadatak Br.3

Program kreiramo tako što ubacujemo dva **ComboBox** elementa u formu i dajemo im imena „ComboBoxFromCurrency“ i „ComboBoxToCurrency“, elementa **Button** kojoj dajemo ime „ButtonConvert“ i tekst prikazan na dugmetu je Konvertuj koji daje indicaciju da nakon njegovog klikljanja će izvršiti konverziju. Također u program ubacujemo dva **TextBox** elementa, u lijevi element unosimo svotu novca koju želimo da konvertujemo, a u desnoj konvertovata valuta će se prikazati nakon klika na „Konvertuj“ dugme.

```
Public Class Form1
    Private tecajDolarEuro As Double = 0.85
    Private tecajEuroDolar As Double = 1.18
    Private tecajDolarKM As Double = 1.68
    Private tecajKMDolar As Double = 0.6
    Private tecajDolarRSD As Double = 102.5
    Private tecajRSDDolar As Double = 0.01

    Private Sub ButtonConvert_Click(sender As Object, e As EventArgs) Handles
ButtonConvert.Click
        Dim iznos As Double
        If Double.TryParse(TextBoxFromAmount.Text, iznos) Then
            Dim polaznaValuta As String = ComboBoxFromCurrency.SelectedItem.ToString()
            Dim ciljanaValuta As String = ComboBoxToCurrency.SelectedItem.ToString()

            Dim konvertovaniIznos As Double = KonvertujValute(polaznaValuta,
ciljanaValuta, iznos)
            TextBoxToAmount.Text = konvertovaniIznos.ToString("0.00")
        Else
```

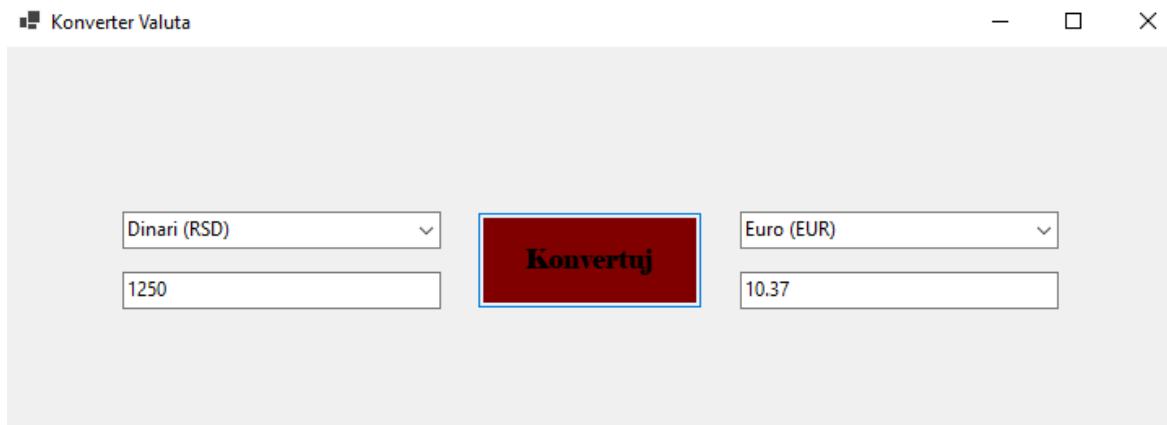
```

        TextBoxToAmount.Text = "Nevažeći iznos."
    End If
End Sub

Private Function KonvertujValute(polaznaValuta As String, ciljanaValuta As String,
iznos As Double) As Double
    Select Case polaznaValuta
        Case "Dolar (USD)"
            Select Case ciljanaValuta
                Case "Euro (EUR)"
                    Return iznos * tecajDolarEuro
                Case "Konvertibilna marka (BAM)"
                    Return iznos * tecajDolarKM
                Case "Dinari (RSD)"
                    Return iznos * tecajDolarRSD
            End Select
        Case "Euro (EUR)"
            Select Case ciljanaValuta
                Case "Dolar (USD)"
                    Return iznos * tecajEuroDolar
                Case "Konvertibilna marka (BAM)"
                    Return iznos * tecajDolarKM / tecajDolarEuro
                Case "Dinari (RSD)"
                    Return iznos * tecajDolarRSD / tecajDolarEuro
            End Select
        Case "Konvertibilna marka (BAM)"
            Select Case ciljanaValuta
                Case "Dolar (USD)"
                    Return iznos * tecajKMDolar
                Case "Euro (EUR)"
                    Return iznos * tecajDolarEuro / tecajDolarKM
                Case "Dinari (RSD)"
                    Return iznos * tecajDolarRSD / tecajDolarKM
            End Select
        Case "Dinari (RSD)"
            Select Case ciljanaValuta
                Case "Dolar (USD)"
                    Return iznos * tecajRSDDolar
                Case "Euro (EUR)"
                    Return iznos * tecajDolarEuro / tecajDolarRSD
                Case "Konvertibilna marka (BAM)"
                    Return iznos * tecajDolarKM / tecajDolarRSD
            End Select
    End Select

    Return 0
End Function
End Class

```



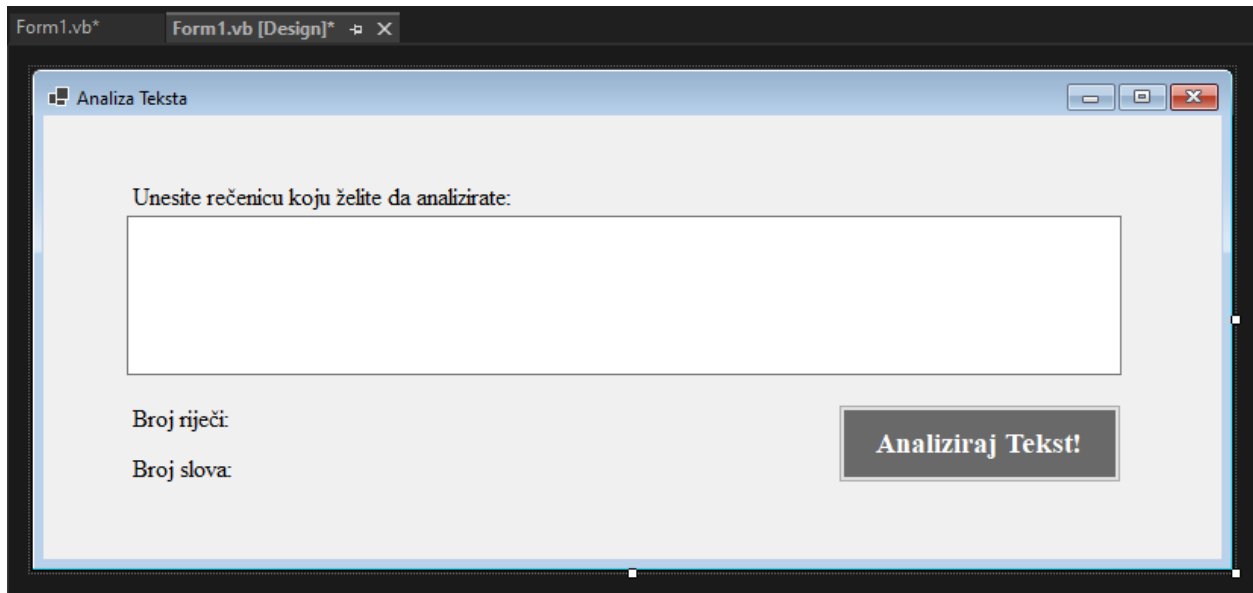
Slika 14 - Rezultat Konvertera Valuta

Na početku koda deklariramo privatne varijable koje sadrže tečajeve (tzv. kurseve) valuta za konverziju. Kada korisnik pritisne dugme „Konvertuj“, poziva se događaj „ButtonConvert_Click“. U ovom dijelu koda, prvo se provjerava da li je korisnik pravilno unio iznos u polju za unos (TextBoxFromAmount). Korištenjem metode **Double.TryParse()** provjeravamo da li je unijeti tekst broj.

Nakon što smo dobili ispravan iznos, izdvajamo selektovane vrijednosti iz **ComboBox**-ova, koje predstavljaju polaznu i ciljanu valutu. Nakon toga, pozivamo funkciju „KonvertujValute“ sa parametrima polazne i ciljane valute, koristeći tečajeve (tzv. kurs) koje smo prethodno definisali i konvertovani iznos se skladišti u varijablu „konvertovaniIznos“.

Na samom kraju, konvertovani iznos se prikazuje u **TextBox**-u sa dve decimale, koristeći formatiranje ToString(0.00). Ako unos valute nije ispravan, tada će TextBox prikazati poruku „Nevažeći iznos.“.

2.2.4. Analiza Teksta



Slika 15 - Prikaz forme u VB Studio Zadatak Br.4

Program započinjemo tako što dodajemo **Label** element na vrhu, gdje od korisnika tražimo da unese tekst tj. sadržaj u **TextBox** element, kojem smo dali ime "TextBoxInput". U programu se također susrećemo sa još dva **Label** elementa kojima smo dali imena „LabelWordCount“ i „LabelCharacterCount“. U programu se nalazi **Button** element sa imenom „ButtonAnalyze“ koji će svojim klikom ažurirati Label Elemente „LabelCharacterCount“ u kom će se prikazati broj slova u unesenom tekstu i „LabelWordCount“ u kom će se prikazati broj riječi.

```
Public Class Form1
    Private Sub ButtonAnalyze_Click(sender As Object, e As EventArgs) Handles
        ButtonAnalyze.Click
            Dim inputText As String = TextBoxInput.Text

            Dim wordCount As Integer = BrojRijeci(inputText)
            LabelWordCount.Text = "Broj riječi: " & wordCount.ToString()

            Dim characterCount As Integer = BrojSlova(inputText)
            LabelCharacterCount.Text = "Broj slova: " & characterCount.ToString()
        End Sub

    Private Function BrojRijeci(text As String) As Integer
```

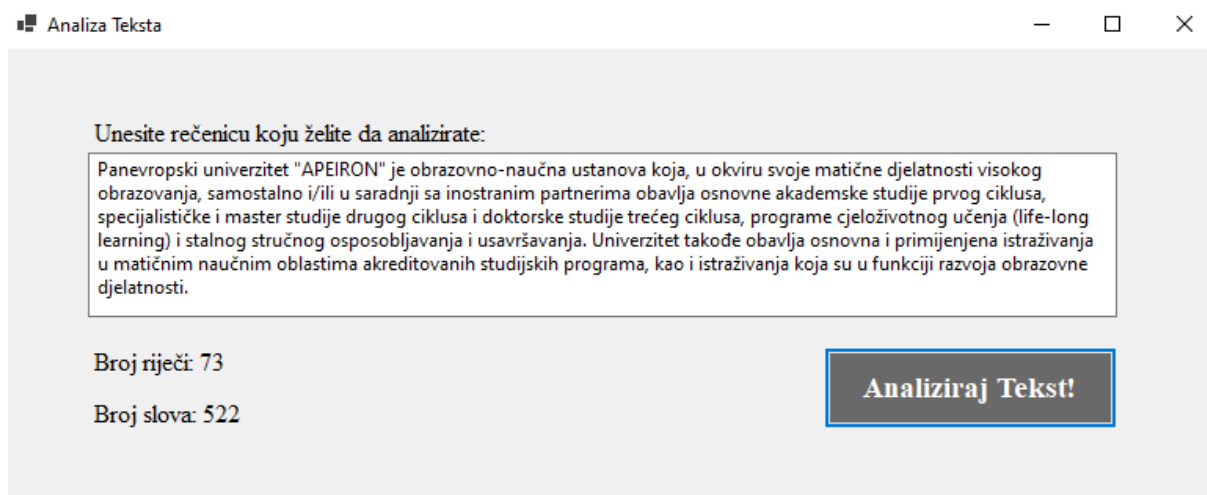
```

    Dim words As String() = text.Split(New Char() {" ", ".", ",", "!", "?", ":", ";", "c"}, StringSplitOptions.RemoveEmptyEntries)
    Return words.Length
End Function

Private Function BrojSlova(text As String) As Integer

    Dim letterCount As Integer = 0
    For Each c As Char In text
        If Char.IsLetter(c) Then
            letterCount += 1
        End If
    Next
    Return letterCount
End Function
End Class

```



Unesite rečenicu koju želite da analizirate:

Panevropski univerzitet "APEIRON" je obrazovno-naučna ustanova koja, u okviru svoje matične djelatnosti visokog obrazovanja, samostalno i/ili u saradnji sa inostranim partnerima obavlja osnovne akademske studije prvog ciklusa, specijalističke i master studije drugog ciklusa i doktorske studije trećeg ciklusa, programe cjeloživotnog učenja (life-long learning) i stalnog stručnog osposobljavanja i usavršavanja. Univerzitet takođe obavlja osnovna i primijenjena istraživanja u matičnim naučnim oblastima akreditovanih studijskih programa, kao i istraživanja koja su u funkciji razvoja obrazovne djelatnosti.

Broj riječi: 73

Broj slova: 522

Analiziraj Tekst!

Slika 16 - Rezultat analize teksta

Na samom početku koda, imamo događaj (engl. event) **ButtonAnalyze_Click** koji se pokreće kada korisnik klikne na dugme „ButtonAnalyze“. U ovom slučaju, prvo se uzima tekst iz TextBox elementa „TextBoxInput“ i smiještamo ga u varijablu „InputText“.

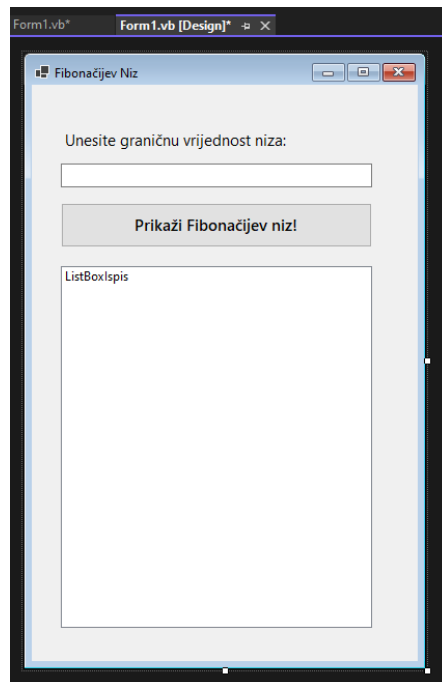
Koristimo funkciju **BrojRijeci** da bi se prebrojale riječi u unešenom tekstu. Ova funkcija će podijeliti tekst na riječi koristeći znake interpunkcije i zatim izračunati broj riječi koja se skladišti u varijabli „wordCount“. Funkcija **BrojSlova** prolazi kroz svaki karakter u tekstu i koristi funkciju **Char.IsLetter** da bi provjerila da li je karakter ili slovo i broj prebrojanih slova se skladišti u varijablu „characterCount“.

2.2.5. Fibonačijev niz

```
Public Class Form1
    Private Sub ButtonPrikazi_Click(sender As Object, e As EventArgs) Handles
        ButtonPrikazi.Click
            Dim clan, i, granica, prviClan, drugiClan As Integer

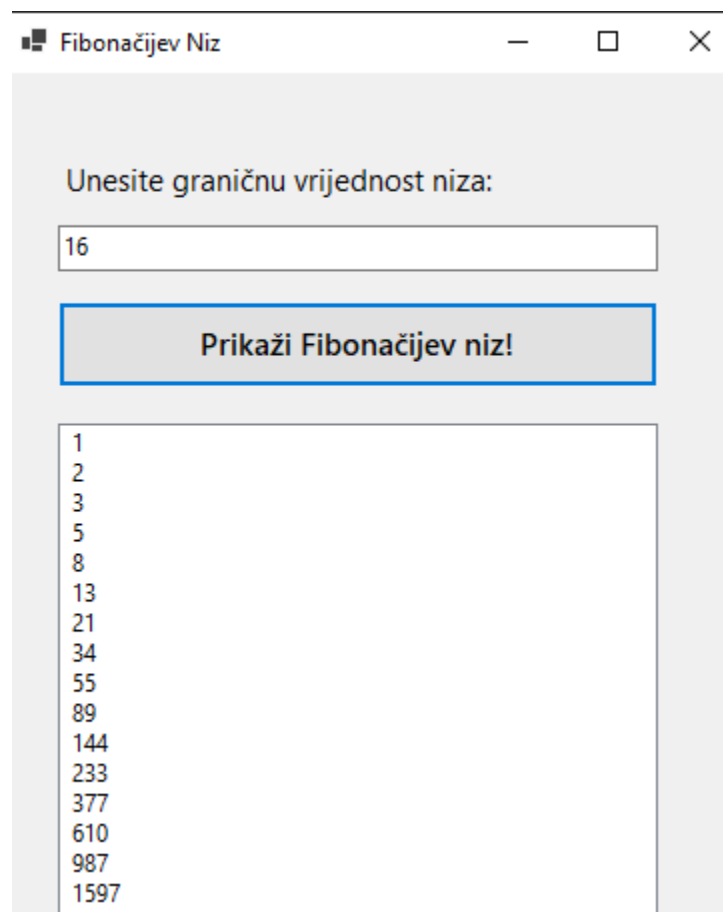
            granica = CInt(TextBoxGranica.Text)
            prviClan = 1
            drugiClan = 2

            If granica < 1 Or granica > 50 Then
                MsgBox("Unesite granicu vrijednost niza koja je veca ili jednaka 1 i manje od
50.")
            ElseIf granica = 1 Then
                ListBoxIspis.Items.Add(Str(prviClan))
            ElseIf granica = 2 Then
                ListBoxIspis.Items.Add(Str(prviClan))
                ListBoxIspis.Items.Add(Str(drugiClan))
            Else
                ListBoxIspis.Items.Add(Str(prviClan))
                ListBoxIspis.Items.Add(Str(drugiClan))
                For i = 3 To granica
                    clan = prviClan + drugiClan
                    ListBoxIspis.Items.Add(Str(clan))
                    prviClan = drugiClan
                    drugiClan = clan
                Next
            End If
        End Sub
End Class
```



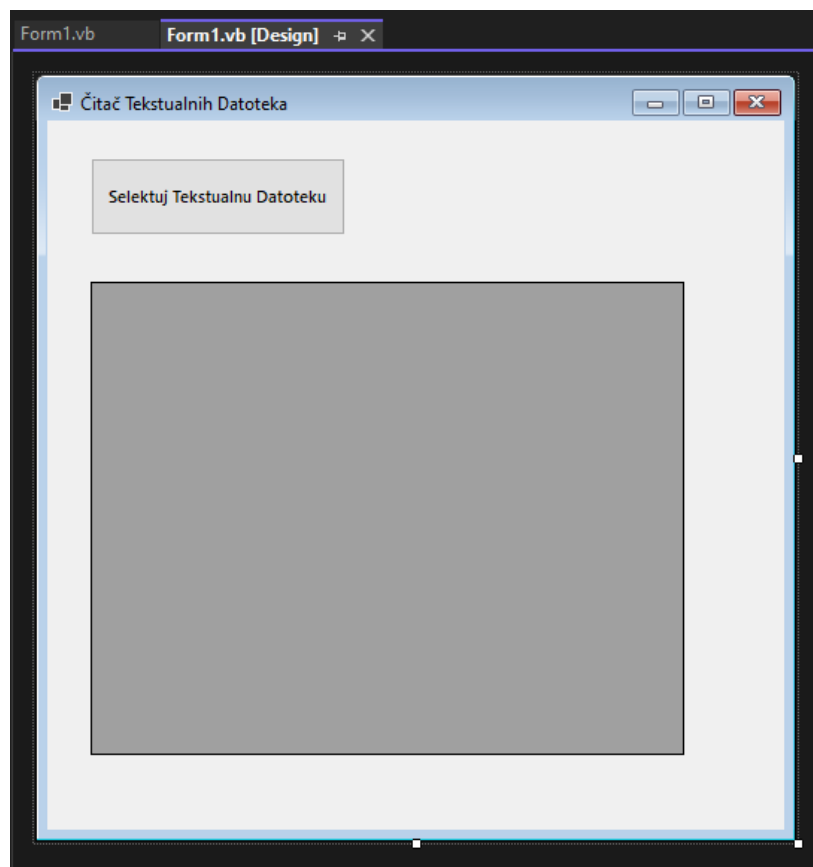
Slika 17 - Prikaz forme u VB Studio Zadatak Br.5

Glavna funkcija ovog programa je **Private Sub ButtonPrikazi_Click(sender As Object, e As EventArgs) Handles ButtonPrikazi.Click**, koja se aktivira kada korisnik pritisne na dugme. Deklarišemo tri varijable **granica, prviClan, drugiClan As Integer** koje koristimo za spremanje unosa granice niza i prva dva člana Fibonačijevog niza. **Granica = CInt(TextBoxGranica.Text)**, koristimo CInt funkciju za pretvaranje unosa iz TextBoxGranica.Text u cijelobrojnu vrijednost i skladištenje vrijednosti u varijablu **granica**. Provjeru granice vrsimo **If granica < 1 Or granica > Then**, ako granica nije u ispravnom rasponu prikazujemo MsgBox sa porukom koja upozorava korisnika da unese ispravnu granicu.



Slika 18 - Rezultat Fibonačijevog niza

2.2.6. Čitač tekstualnih datoteka



Slika 19 - Prikaz forme u VB Studio zadatak Br.6

U ovom programu se koriste tri elementa od kojih su Button, DataGridView i OpenFileDialog. Elementu Button dajemo ime „ButtonOdaberiDatoteku“ koju koristimo za odabir datoteke. Korisnik može kliknuti na ovo dugme kako bi otvorio prozor za odabir datoteke **OpenFileDialog**. Kada korisnik odabere datoteku njena putanja (engl. path) se skladišti u varijablu „filePath“. DataGridView element predstavlja tabelarni prikaz podataka.

Prvo se čiste prethodno prikazani podaci, a potom ručno definišemo tri kolone koje ćemo prikazati „Imena Studenata“, „Prezimana Studenata“ i „Indeksi Studenata“. Učitavamo sadržaj tekstualne datoteke u niz **lines** pomoću metode **File.ReadAllLines(filePath)**, nakon toga koristimo petlju **For Each** kako bi prošli kroz svaki red učitanoog teksta.

Svaki red dijelimo na tri dijela, koristeći metodu **Split(",")** jer su podaci odvojeni zarezom. Na samom kraju programa koristimo metodu **dataGridView.Rows.Add(cells)** za dodavanje kolona u DataGridView. Svaki element u nizu **cells** predstavlja jednu kolonu u DataGridView-u.

```
Imports System.IO
```

```
Public Class Form1
```

```
Private Sub ButtonOdaberiDatoteku_Click(sender As Object, e As EventArgs) Handles  
ButtonOdaberiDatoteku.Click
```

```
Dim openFileDialog As New OpenFileDialog()  
openFileDialog.Filter = "Tekstualne datoteke (*.txt)|*.txt|Sve datoteke  
(*.*)|*.*"
```

```
openFileDialog.FilterIndex = 1  
openFileDialog.RestoreDirectory = True  
If openFileDialog.ShowDialog() = DialogResult.OK Then  
Dim filePath As String = openFileDialog.FileName  
UcitajPodatke(filePath)
```

```
End If
```

```
End Sub
```

```
Private Sub UcitajPodatke(filePath As String)
```

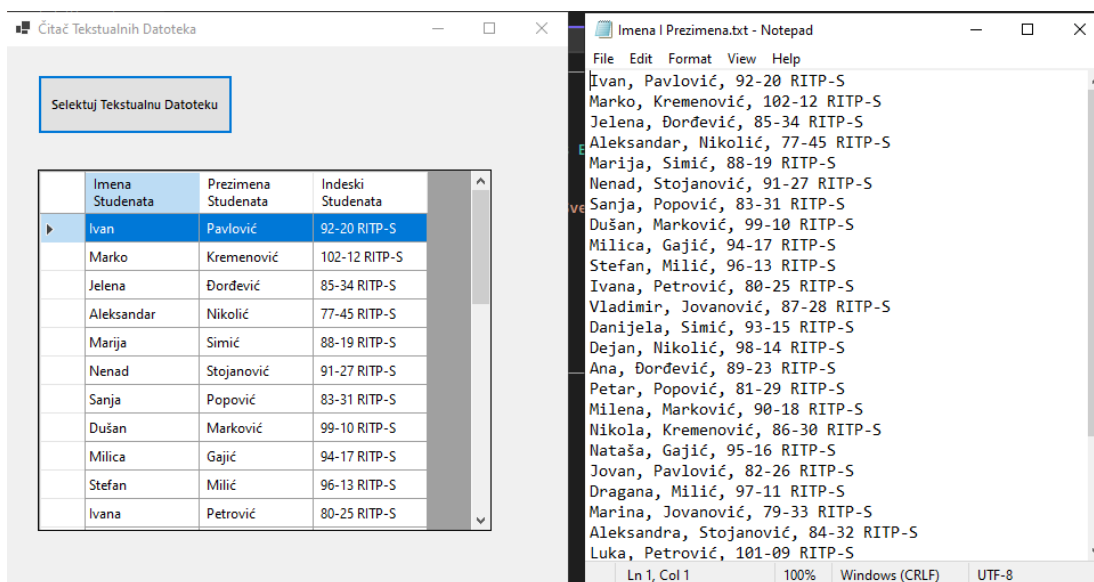
```
dataGridView.Rows.Clear()  
dataGridView.Columns.Clear()  
dataGridView.Columns.Add("Column1", "Imena Studenata")  
dataGridView.Columns.Add("Column2", "Prezimana Studenata")  
dataGridView.Columns.Add("Column3", "Indeksi Studenata")  
Dim lines As String() = File.ReadAllLines(filePath)
```

```
For Each line As String In lines  
Dim cells As String() = line.Split(",")  
dataGridView.Rows.Add(cells)
```

```
Next
```

```
End Sub
```

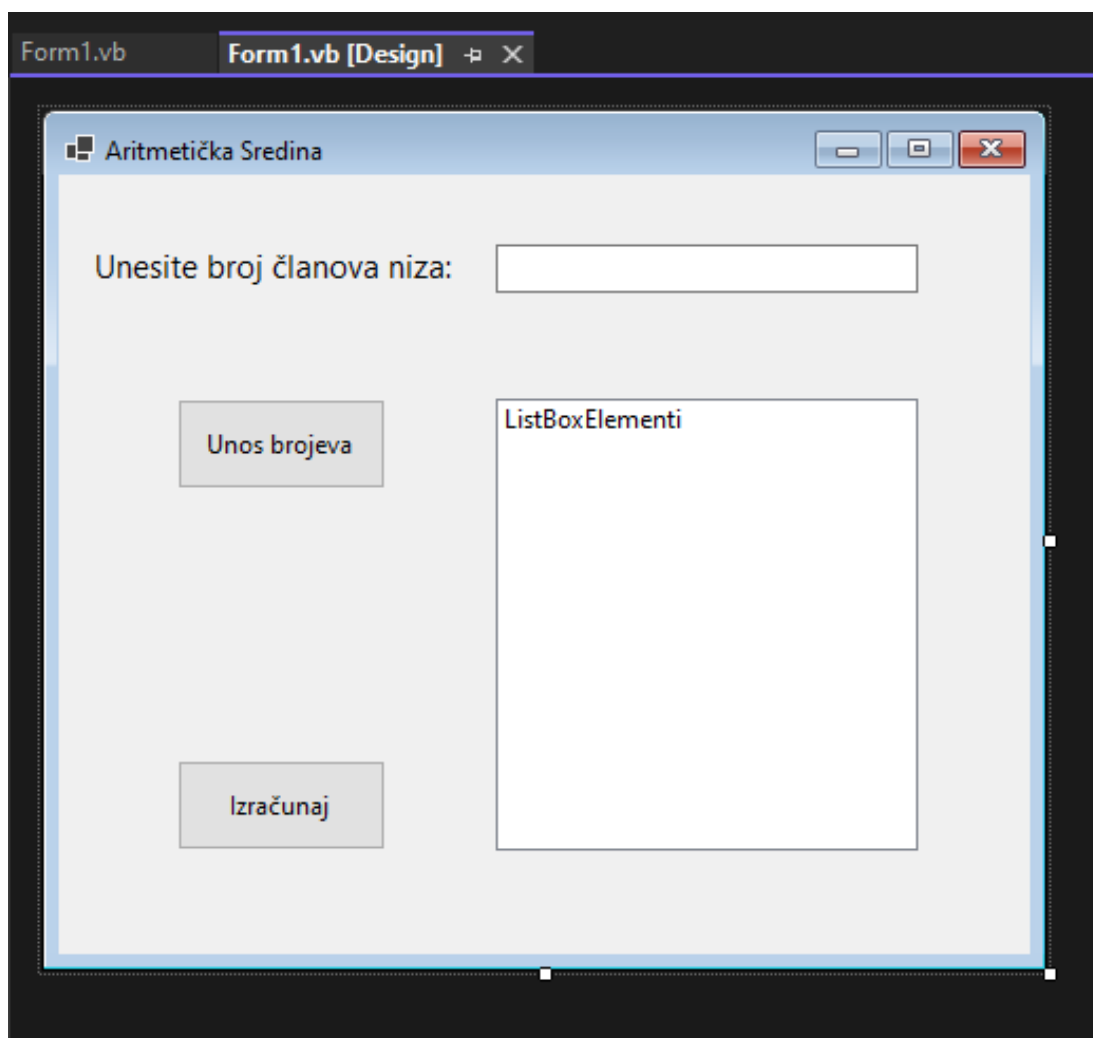
```
End Class
```



Slika 20 - Rezultat Čitača Tekstualnih Datoteka

2.2.7. Aritmetička sredina

Program za izračunavanje Aritmetičke sredine počinjemo tako što korisnik unosi broj elemenata niza preko TextBox-a „txtBrojElemenata“ i kad su elementi dodati, korisnik pritišće dugme „btnElementi“. Program otvara InputBox za unos broja za svaki element niza, zatim provjerava da li su unešeni brojevi validni. Ako korisnik unese neki karakter da nije broj, prikazaće se poruka o grešci. Nakon unosa svih elemenata niza, korisnik može pritisnuti na „btnIzracunaj“. Nakon toga, program izračunava aritmetičku sredinu unesenih brojeva i prikazuje rezultat u MessageBox-u.



Slika 21 - Prikaz forme u VB Studio zadatak Br.7

```

Public Class Form1
    Dim n As Integer
    Dim niz(100) As Double
    Private Sub btnElementi_Click(sender As Object, e As EventArgs) Handles
        btnElementi.Click
            On Error GoTo greska
            n = CInt(txtBrojElemenata.Text)

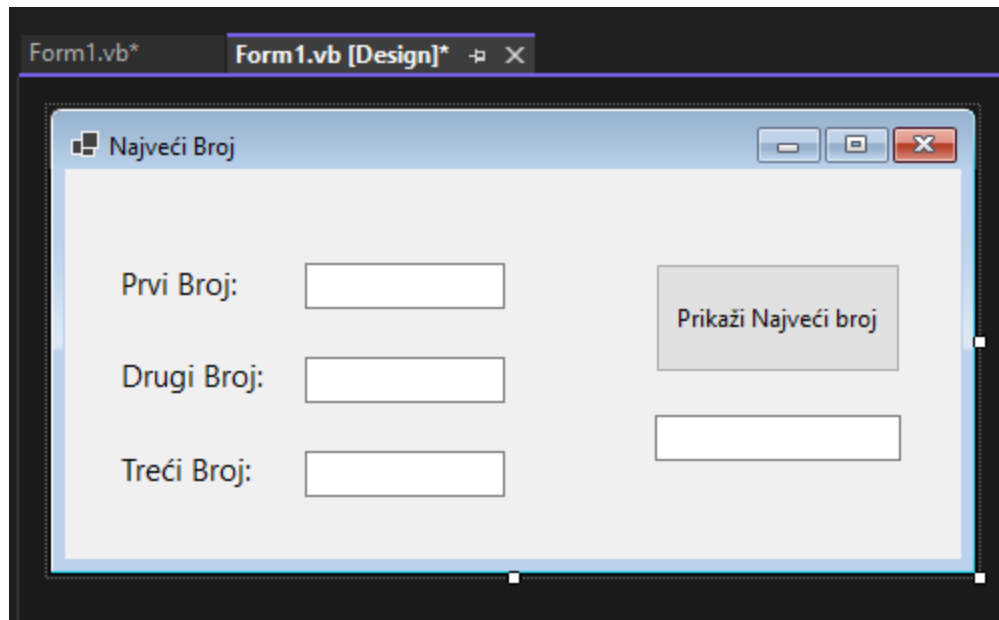
            If n <= 0 Then
                Throw New Exception("Broj elemenata mora biti pozitivan broj.")
            End If

            ListBoxElementi.Items.Clear()
            For i = 1 To n
                Dim uneseniBroj As String = InputBox("Unesite " & i & ". element niza")
                Dim broj As Double
                If Not Double.TryParse(uneseniBroj, broj) Then
                    Throw New Exception("Vjerovatno ste pogriješili tokom unosa, unoseći znakove
umjesto brojeva.")
                End If
                niz(i) = broj
                ListBoxElementi.Items.Add(broj)
            Next
        Exit Sub
    greska:
        MessageBox.Show("Greška: " & Err.Description,
                        "Pogreška", MessageBoxButtons.OK, MessageBoxIcon.Warning
                        )
        ListBoxElementi.Items.Clear()
    End Sub
    Private Sub btnIzracunaj_Click(sender As Object, e As EventArgs) Handles
        btnIzracunaj.Click
            Dim suma As Double = 0
            For i = 1 To n
                suma += niz(i)
            Next
            Dim aritmetickaSredina As Double = suma / n
            MessageBox.Show("Aritmetička sredina je: " & aritmetickaSredina.ToString(),
                            "Rezultat", MessageBoxButtons.OK, MessageBoxIcon.Information )
        End Sub
    End Class

```

2.2.8. Najveći broj od unesenih brojeva

U programu za izračunavanje najvećeg unesenog broja koristimo varijable **a**, **b** i **c** u koje skladištimo unesene brojeve. Koristimo **Integer.TryParse()** metodu da pretvorimo uneseni tekst u cijelobrojan broj. Ako unos nije validan, varijabla će ostati nula, što nam pomaže u provjeri validnosti unosa. Varijabla najveći sadrži najveći broj od unesena tri broja. Koristimo **Math.Max()** funkciju koja upoređuje tri broja i vraća najveći od njih.



Slika 22 - Prikaz forme u VB Studio zadatak Br.8

```
Public Class Form1
    Private Sub btnIzracunaj_Click(sender As Object, e As EventArgs) Handles
        btnIzracunaj.Click
            Dim a, b, c As Integer

            Integer.TryParse(txtPrvi.Text, a)
            Integer.TryParse(txtDrugi.Text, b)
            Integer.TryParse(txtTreci.Text, c)

            Dim najveci As Integer = Math.Max(a, Math.Max(b, c))

            If a = b And b = c Then
                txtNajveci.Text = "Brojevi su jednaki!"
            Else
                txtNajveci.Text = najveci.ToString()
            End If
        End Sub
    End Class
```

2.2.9. Jednostavna Paint alatka

Ovaj program koristi funkcionalnosti WinForms-a kako bi omogućio korisniku crtanje s bojama i također njegovo brisanje. Klikom na **PictureBox** i povlačenjem miša (dok je pritisnuto lijevo dugme miša), korisnik može crtati linije. Boja se mijenja odabirom jednom od dugmadi za boje. Klikom na dugme „Gumica“, boja za crtanje postaje bijela boja što omogućuje korisniku brisanje crteža tako što će ga jednostavno prekriti bijelom bojom.



Slika 23- Prikaz forme u VB Studio zadatak Br.9

```
Public Class Form1
    Dim dole As Boolean = False
    Dim mojaBoja As Brush = Brushes.Black
    Private Sub PictureBox1_MouseDown(sender As Object, e As MouseEventArgs) Handles
PictureBox1.MouseDown
        dole = True
    End Sub
    Private Sub PictureBox1_MouseUp(sender As Object, e As MouseEventArgs) Handles
PictureBox1.MouseUp
        dole = False
    End Sub
    Private Sub PictureBox1_MouseMove(sender As Object, e As MouseEventArgs) Handles
PictureBox1.MouseMove
        If dole Then
            Dim g As Graphics = PictureBox1.CreateGraphics()
            g.FillEllipse(mojaBoja, e.X, e.Y, 10, 10)
            g.Dispose()
        End If
    End Sub
    Private Sub btnBoja_Click(sender As Object, e As EventArgs) Handles btnCrna.Click,
btnPlava.Click, btnCrvena.Click, btnZelena.Click, btnZuta.Click, btnSiva.Click,
btnSmedja.Click
        Dim btn As Button = CType(sender, Button)
        Select Case btn.Name
            Case "btnCrna"
                mojaBoja = Brushes.Black
            Case "btnPlava"
                mojaBoja = Brushes.Blue
            Case "btnCrvena"

```

```

        mojaBoja = Brushes.Red
    Case "btnZelena"
        mojaBoja = Brushes.Green
    Case "btnZuta"
        mojaBoja = Brushes.Yellow
    Case "btnSiva"
        mojaBoja = Brushes.Gray
    Case "btnSmedja"
        mojaBoja = Brushes.Brown
End Select
End Sub
Private Sub btnBrisanje_Click(sender As Object, e As EventArgs) Handles
btnBrisanje.Click
    mojaBoja = Brushes.White
End Sub
End Class

```

2.2.10. Izračun mjesečne rate kredita

Ovaj program služi za izračunavanje mjesečne rate kredita na osnovu unesenih podataka o iznosu kredita, broju rata i kamatnoj stopi. Od korisnika se traži da unese Iznos kredita, Broj rata i Kamatnu stopu. Kada korisnik pritisne dugme “btnIzracunaj”, program prvo provjerava da li su uneseni podaci tačni tj. validni. Ako su uneseni podaci tačni, program izračunava mjesečnu ratu kredita koristeći ugrađenu funkciju **Pmt** koja radi sa finansijskim funkcijama u Microsoft Excel. Rezultat tj. Mjesečna rata se prikazuje u TextBox-u “txtMjesecnaRata”, zaokružena sa dve decimale.

The screenshot shows a Windows Form titled "Form1" with a light gray background. It contains the following elements:

- A label "Kredit:" followed by a text box containing the value "2241".
- A label "Broj rata:" followed by a text box containing the value "12".
- A label "Kamatna stopa:" followed by a text box containing the value "3".
- A rectangular button with a blue border and the text "Izračunaj" in the center.
- A label "Mjesečna Rata:" followed by a text box containing the value "189.8".

Slika 24 - Prikaz forme u VB Studio zadatak Br.10

```

Public Class Form1
    Private Sub BtnIzracunaj_Click(sender As Object, e As EventArgs) Handles
btnIzracunaj.Click
        Dim kredit As Double
        Dim brojRata As Integer
        Dim kamatnaStopa As Double
        Dim mesecnaRata As Double
        If Not Double.TryParse(txtKredit.Text, kredit) OrElse kredit <= 0 Then
            MessageBox.Show("Unesite ispravan iznos kredita.", "Greška",
MessageBoxButtons.OK, MessageBoxIcon.Warning)
            Return
        End If
        If Not Integer.TryParse(txtBrojRata.Text, brojRata) OrElse brojRata <= 0 Then
            MessageBox.Show("Unesite ispravan broj rata.", "Greška", MessageBoxButtons.OK,
MessageBoxIcon.Warning)
            Return
        End If

        If Not Double.TryParse(txtKamatnaStopa.Text, kamatnaStopa) OrElse kamatnaStopa
<= 0 Then
            MessageBox.Show("Unesite ispravnu kamatnu stopu.", "Greška",
MessageBoxButtons.OK, MessageBoxIcon.Warning)
            Return
        End If
        kamatnaStopa = kamatnaStopa / 100 / 12
        mesecnaRata = Pmt(kamatnaStopa, brojRata, -kredit)
        txtMjesecnaRata.Text = Math.Round(mesecnaRata, 2).ToString()
    End Sub
End Class

```

2.2.11. Bubble Sort

Program koji implementira algoritam Bubble Sort se izvršava tako što korisnik unosi broj članova niza, klikom na dugme „btnUcitaj“, nakon toga se otvara InputBox u kom korisnik unosi željeni broj članova. Nakon unosa broja članova, korisniku se prikazuje broj InputBox-ova u kojima može unijeti članove niza jedan po jedan. Unesene vrijednosti se spremaju u varijablu „niz“. Kada korisnik završi unos članova niza, pritiskanjem na dugme „btnSortiraj“, niz se sortira koristeći Bubble Sort algoritam. Bubble Sort niz radi tako što prolazi kroz niz više puta i u svakom prolazu, upoređuje članove niza i ako je potrebno, zamijenjuje ih tako da se najveći članovi pomjere prema kraju niza. Sortirani niz se prikazuje u drugom ListBox-u.

```

Public Class Form1
    Dim niz(100) As Integer
    Dim broj As Integer

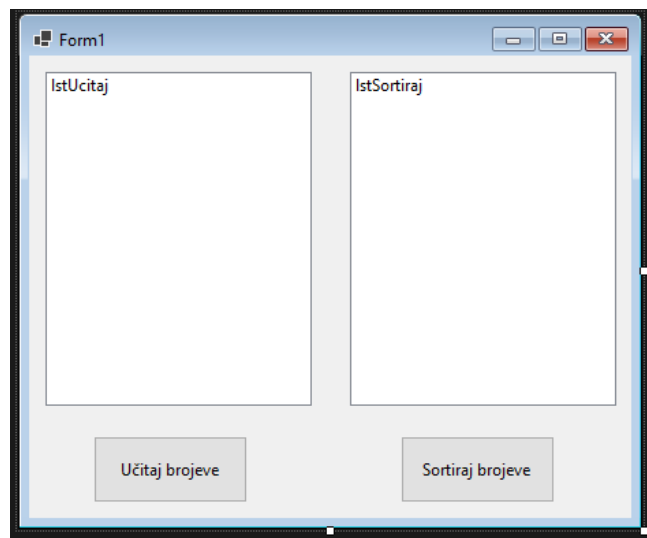
    Private Sub btnUcitaj_Click(sender As Object, e As EventArgs) Handles
btnUcitaj.Click
        lstUcitaj.Items.Clear()

```

```

    lstSortiraj.Items.Clear()
    broj = InputBox("Unesi broj članova niza.")
    Unesi(broj)
    Ispisi(lstUcitaj)
End Sub
Private Sub Unesi(ByVal broj)
    For i = 1 To broj
        niz(i) = InputBox("Unesite " & i & ". element niza.")
    Next
End Sub
Private Sub Ispisi(ByVal listBoxNiz As ListBox)
    For i = 1 To broj
        listBoxNiz.Items.Add(niz(i))
    Next
End Sub
Private Sub Sortiraj(ByVal nizz As Integer())
    For i = 1 To broj
        For j = 1 To broj - 1
            If nizz(j) >= nizz(j + 1) Then
                Zamijeni(nizz, j, j + 1)
            End If
        Next
    Next
End Sub
Private Sub Zamijeni(ByRef nizz As Integer(), ByVal index1 As Integer, ByVal
index2 As Integer)
    Dim pom As Integer
    pom = nizz(index1)
    nizz(index1) = nizz(index2)
    nizz(index2) = pom
End Sub
Private Sub btnSortiraj_Click(sender As Object, e As EventArgs) Handles
btnSortiraj.Click
    Sortiraj(niz)
    Ispisi(lstSortiraj)
End Sub
End Class

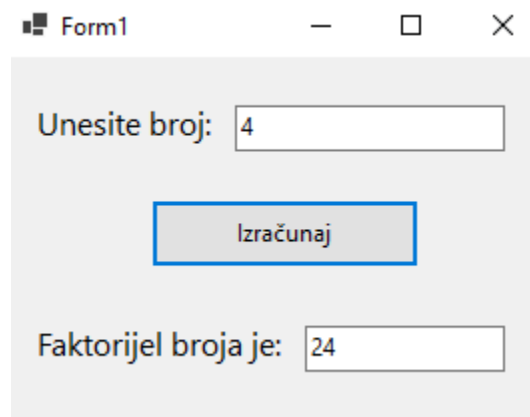
```



Slika 25- Prikaz forme u VB Studio zadatak Br.11

2.2.12. Faktorijel broja

U ovom programu korisnik izračunava faktorijel prirodnog broja tako što korisnik unosi prirodan broj u TextBox „txtUneseniBroj“. Kada korisnik pritisne dugme „btnIzracunaj“, program prvo provjerava da li su unešeni podaci validni. Kada je unos validan, program koristi rekurzivnu metodu „IzracunajFaktorijel“ za izračunavanje faktorijela unešenog broja.



Slika 26 - Prikaz forme u VB Studio zadatak Br.12

```
Public Class Form1
    Private Sub btnIzracunaj_Click(sender As Object, e As EventArgs) Handles
        btnIzracunaj.Click
            Dim broj As Integer
            Dim faktorijel As Long

            If Not Integer.TryParse(txtUneseniBroj.Text, broj) OrElse broj < 0 Then
                MessageBox.Show("Unesite prirodan broj.", "Greška", MessageBoxButtons.OK,
                MessageBoxIcon.Error)
                Return
            End If

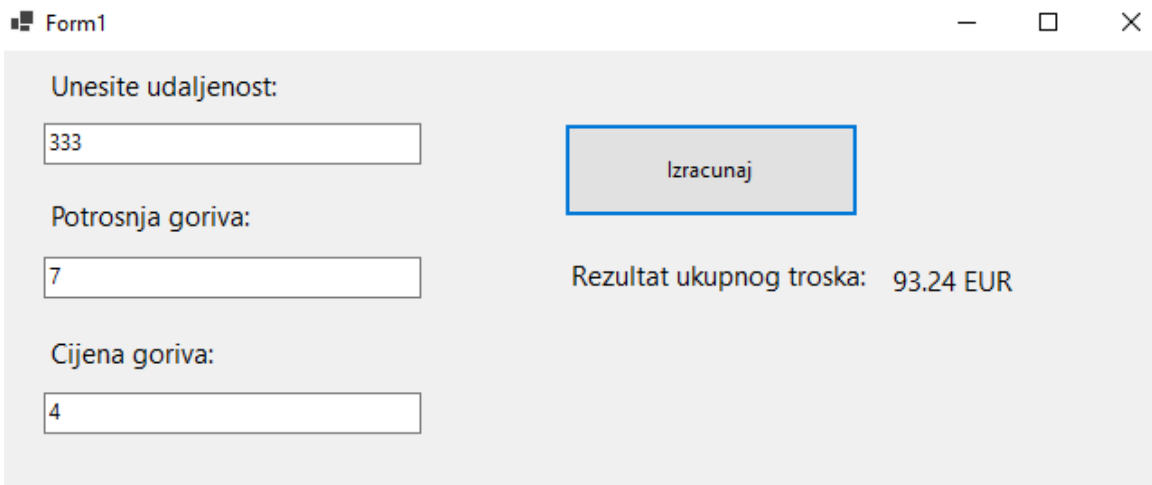
            faktorijel = IzracunajFaktorijel(broj)

            txtFaktorijel.Text = faktorijel.ToString()
        End Sub

        Private Function IzracunajFaktorijel(ByVal n As Integer) As Long
            If n = 0 Then
                Return 1
            Else
                Return n * IzracunajFaktorijel(n - 1)
            End If
        End Function
    End Class
```

2.2.13. Kalkulator potrošnje putovanja

Ova aplikacija omogućava korisnicima da izračunaju troškove tako što korisnik unosi udaljenost putovanja u kilometrima, potrošnju goriva vozila u litrama na 100 kilometara i cijenu goriva po litri. Nakon pritiska na dugme „Izračunaj“, program automatski provjerava tačnost unesenih podataka. Ako su svi podaci ispravno uneseni, koristi ih da izračun ukupnih troškova putovanja. Rezultat se prikazuje na ekranu u vidu teksta koji pokazuje ukupan iznos troškova putovanja. Ova aplikacija olakšava korisnicima brz i precizan izračun troškova njihovih putovanja, čime im pomaže da planiraju budžet i upravljaju svojim troškovima tokom puta.



Slika 27 - Prikaz forme u VB Studio zadatak Br.13

```
Public Class Form1
    Private Sub ButtonCalculate_Click(sender As Object, e As EventArgs) Handles
        ButtonCalculate.Click
            If String.IsNullOrEmpty(TextBoxDistance.Text) OrElse
                String.IsNullOrEmpty(TextBoxFuelConsumption.Text) OrElse
                String.IsNullOrEmpty(TextBoxFuelPrice.Text) Then
                MessageBox.Show("Molimo unesite sve potrebne vrijednosti.", "Greška",
                    MessageBoxButtons.OK, MessageBoxIcon.Error)
                Return
            End If

            Dim distance As Double
            Dim fuelConsumption As Double
            Dim fuelPrice As Double

            If Not Double.TryParse(TextBoxDistance.Text, distance) OrElse
                Not Double.TryParse(TextBoxFuelConsumption.Text, fuelConsumption) OrElse
                Not Double.TryParse(TextBoxFuelPrice.Text, fuelPrice) Then
```

```

        MessageBox.Show("Neispravan unos. Molimo unesite brojčane vrijednosti.",
"Greška", MessageBoxButtons.OK, MessageBoxIcon.Error)
    Return
End If

    Dim totalCost As Double = (distance / 100) * fuelConsumption * fuelPrice
    LabelTotalCost.Text = $"{totalCost:F2} EUR"
End Sub
End Class

```

2.2.14. Kalkulator brzine tipkanja

Ovaj program ima za cilj mjerenje brzine kucanja korisnika unutar određenog vremenskog intervala. Korisnik započinje vježbu pritiskom na dugme „START“. Nakon toga, tekst koji treba otkucati se prikazuje na ekranu. Korisnik unosi taj tekst u polje za unos teksta, a kada korisnik završi unos pritiskom na taster „Enter“, program izračunava brzinu kucanja u karakterima po minuti na osnovu vremena koje je korisnik potrošio i ukupnih broja unesenih karaktera.



Slika 28 - Prikaz forme u VB Studio zadatak Br.14

```

Public Class Form1
    Dim pocetnoVreme As DateTime
    Dim krajnjeVreme As DateTime
    Dim unosZavršen As Boolean = False
    Private Sub ButtonStart_Click(sender As Object, e As EventArgs) Handles
ButtonStart.Click
        LabelRezultat.Text = "Pritisnite Enter kada završite..."
    End Sub
End Class

```



```

LabelTekst.Text = ""
TextBoxUnos.Clear()
TextBoxUnos.Enabled = True
ButtonStart.Enabled = False
unosZavršen = False
pocetnoVreme = DateTime.Now
End Sub
Private Sub TextBoxUnos_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
TextBoxUnos.KeyPress
    If e.KeyChar = ChrW(Keys.Enter) Then
        If Not unosZavršen Then
            krajnjeVreme = DateTime.Now
            Dim ukupnoSekundi As Double = (krajnjeVreme - pocetnoVreme).TotalSeconds
            Dim unesenoSlova As Integer = TextBoxUnos.Text.Length
            Dim brzinaTipkanja As Integer = CInt(unesenoSlova / ukupnoSekundi * 60)
            LabelRezultat.Text = $"Brzina tipkanja: {brzinaTipkanja} karaktera u minuti"
            TextBoxUnos.Enabled = False
            ButtonStart.Enabled = True
            unosZavršen = True
        End If
    End If
End Sub
End Class

```

2.2.15. Generator lozinke

U programu za generisanje lozinke, samom korisniku se nudi da odabere dužinu svoje lozinke i njene postavke kao što su velika slova, mala slova, brojevi i specijalni karakteri.

Slika 29 - Prikaz forme u VB Studio zadatak Br.15

```

Imports System.Text
Imports System.Security.Cryptography

Public Class Form1
    Private Sub ButtonGeneriši_Click(sender As Object, e As EventArgs) Handles
ButtonGeneriši.Click
        Dim dužinaLozinke As Integer = CInt(NumericUpDownDužina.Value)
        Dim koristiMalaSlova As Boolean = CheckBoxMalaSlova.Checked
        Dim koristiVelikaSlova As Boolean = CheckBoxVelikaSlova.Checked
        Dim koristiBrojeve As Boolean = CheckBoxBrojevi.Checked
        Dim koristiSpecijalneKaraktere As Boolean = CheckBoxSpecijalniKarakteristi.Checked

        Dim dostupniKarakteristi As String = ""
        If koristiMalaSlova Then
            dostupniKarakteristi += "abcdefghijklmnopqrstuvwxyz"
        End If
        If koristiVelikaSlova Then
            dostupniKarakteristi += "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
        End If
        If koristiBrojeve Then
            dostupniKarakteristi += "0123456789"
        End If
        If koristiSpecijalneKaraktere Then
            dostupniKarakteristi += "!@#$%^&*()_-=<>?/"
        End If
        If dostupniKarakteristi.Length = 0 Then
            MessageBox.Show("Morate izabrati bar jednu opciju za generisanje lozinke.",
"Upozorenje", MessageBoxButtons.OK, MessageBoxIcon.Warning)
            Return
        End If

        Dim random As New RNGCryptoServiceProvider()
        Dim randomBytes(dužinaLozinke - 1) As Byte
        random.GetBytes(randomBytes)

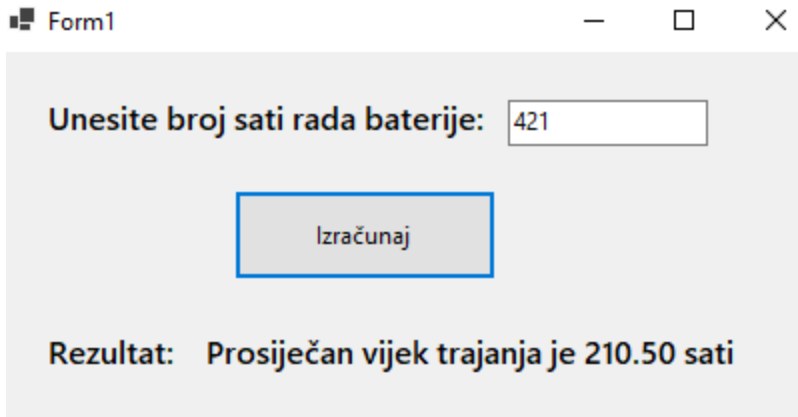
        Dim lozinka As New StringBuilder(dužinaLozinke)
        For Each b As Byte In randomBytes
            lozinka.Append(dostupniKarakteristi(b Mod dostupniKarakteristi.Length))
        Next

        TextBoxLozinka.Text = lozinka.ToString()
    End Sub
End Class

```

2.2.16. Kalkulator životnog vijeka baterije

Ovaj program omogućava korisnicima da unesu broj sati rada baterije i izračuna prosječan vijek trajanja. Nakon unosa broja sati i pritiska dugmeta „Izračunaj“, program prikazuje koliko bi prosječno trajala ta baterija na osnovu unešenih sati.



Slika 30 - Prikaz forme u VB Studio zadatak Br.16

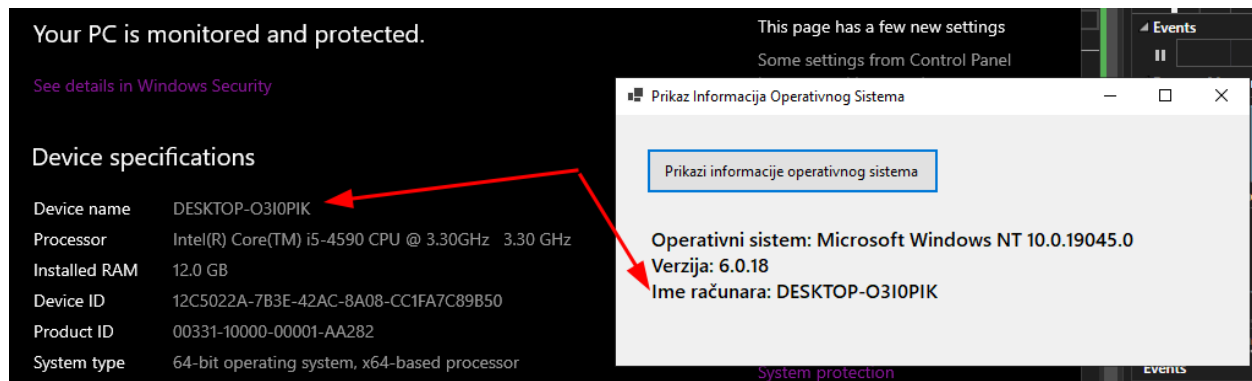
```
Public Class Form1
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load

    End Sub

    Private Sub btnIzracunaj_Click(sender As Object, e As EventArgs) Handles
btnIzracunaj.Click
        Dim hours As Integer
        If Integer.TryParse(txtBrojSati.Text, hours) Then
            Dim averageLife As Double = hours / 2.0
            lblProsecanVekTrajanja.Text = "Prosiječan vijek trajanja je " &
averageLife.ToString("0.00") & " sati"
        Else
            MessageBox.Show("Molimo vas da unesete validan broj sati.")
        End If
    End Sub
End Class
```

2.2.17. Prikaz specifikacija operativnog sistema

Ovaj program omogućava korisniku da klikom na dugme „Prikazi informacije operativnog sistema“ dobiju osnovne informacije o trenutnom operativnom sistemu na svom računaru. Nakon klika na dugme, program prikazuje detalje kao što su ime operativnog sistema, njegova verzija i ime računara na kojem je program pokrenut.



Slika 31 - Prikaz forme u VB Studio zadatak Br.17

```
Public Class Form1
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load

    End Sub

    Private Sub btnPrikaziInformacije_Click(sender As Object, e As EventArgs) Handles
btnPrikaziInformacije.Click

        Dim osInfo As String = "Operativni sistem: " & Environment.OSVersion.ToString()
        osInfo &= vbCrLf & "Verzija: " & Environment.Version.ToString()
        osInfo &= vbCrLf & "Ime računara: " & Environment.MachineName

        lblInformacijeOS.Text = osInfo
    End Sub
End Class
```

2.2.18. Prikaz teksta u različitim bojama

Ovaj program omogućava korisniku da unese tekstualni sadržaj u TextBox, izabere boju iz ComboBox elementa i da prikaže taj tekstualni sadržaj u odabranoj boji pomoću Label elementa. Korištenjem RGB vrijednosti boja, program interpretira izabranu boju i primenjuje je na prikazani tekst.

```
Public Class Form1
    Private Sub btnPrikaziTekst_Click(sender As Object, e As EventArgs) Handles
btnPrikaziTekst.Click
        Dim unetiTekst As String = txtUnetiTekst.Text
        Dim izabranaBoja As Color

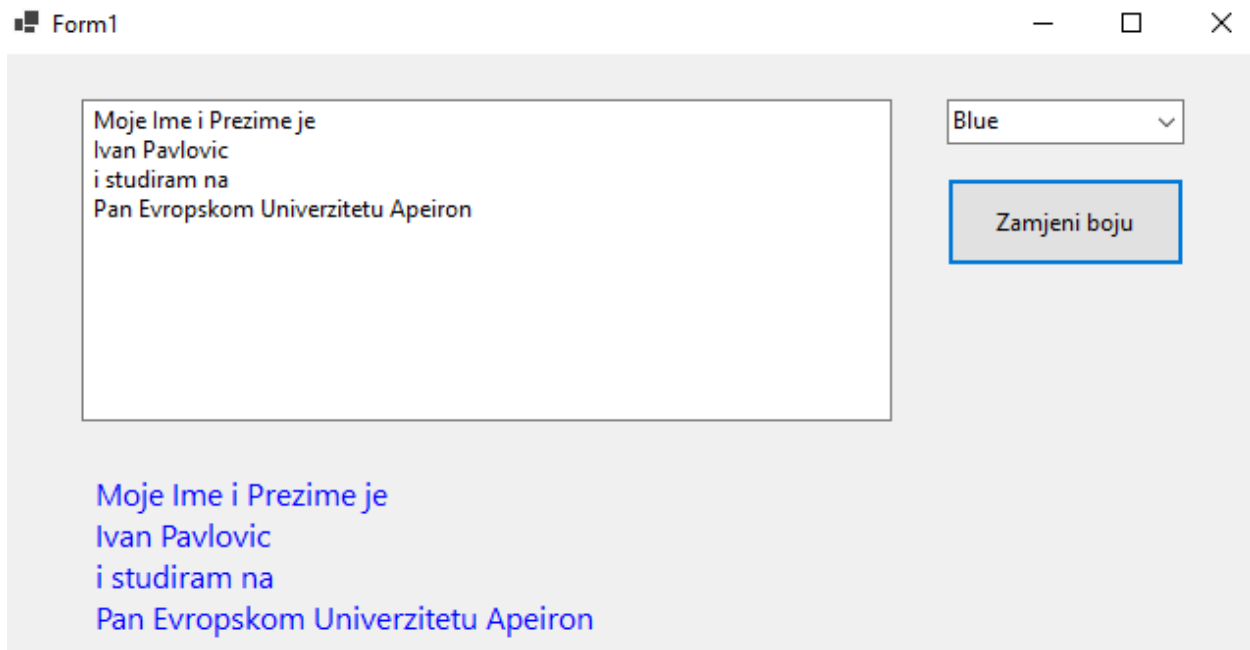
        Select Case cmbIzabranaBoja.SelectedIndex
            Case 0
                izabranaBoja = Color.Red
            Case 1
                izabranaBoja = Color.Green
        End Select
    End Sub
End Class
```

```

Case 2
    izabranaBoja = Color.Blue
Case 3
    izabranaBoja = Color.Yellow
Case Else
    izabranaBoja = Color.Black
End Select

lblPrikazTeksta.ForeColor = izabranaBoja
lblPrikazTeksta.Text = unetiTekst
End Sub
End Class

```

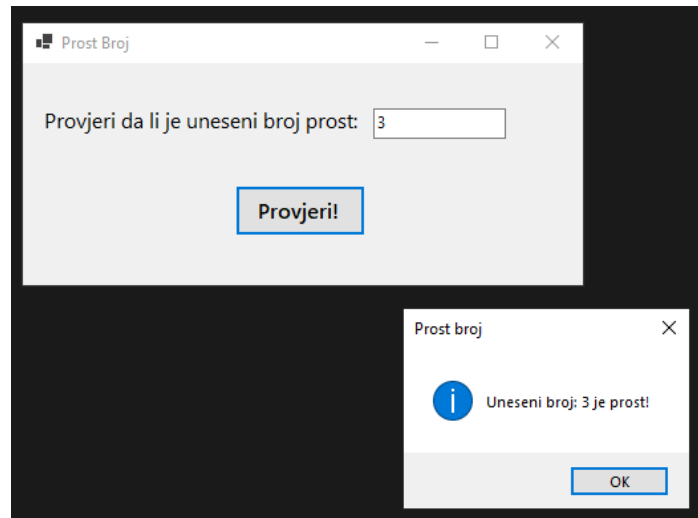


Slika 32 - Prikaz forme u VB Studio zadatak Br.18

2.2.19. Provjera da li je broj prost

Ovaj program omogućava korisniku da unese broj, a zatim provjerava da li je taj broj prost ili nije. Prost broj je broj koji se može podijeliti samo sa samim sobom i brojem 1, dok se ya ostale brojeve može naći više pozitivnih dijelitelja.

Nakon unosa broja, program izračunava sve njegove dijelitelje i ako pronade više od dva, zaključuje da broj nije prost. U suprotnom, ako broj ima samo dva pozitivna dijelitelja, program će zaključiti da je broj prost i prikazat će odgovarajuću poruku korisniku.



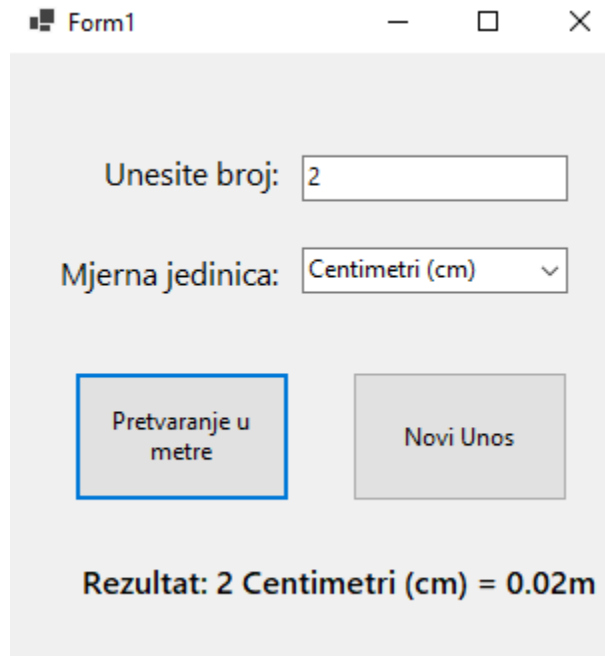
Slika 33 - Prikaz forme u VB Studio zadatak Br.19

```
Public Class Form1
    Private Sub btnTestiraj_Click(sender As Object, e As EventArgs) Handles
        btnTestiraj.Click
            Dim broj As Integer
            If Integer.TryParse(txtUneseniBroj.Text, broj) Then
                If broj <= 1 Then
                    MessageBox.Show("Unesite broj koji je veći od 1.", "Greška",
                        MessageBoxButtons.OK, MessageBoxIcon.Error)
                Else
                    Dim prost As Boolean = JeProstBroj(broj)
                    If prost Then
                        MessageBox.Show($"Uneseni broj: {broj} je prost!", "Prost broj",
                            MessageBoxButtons.OK, MessageBoxIcon.Information)
                    Else
                        MessageBox.Show($"Uneseni broj: {broj} nije prost!", "",
                            MessageBoxButtons.OK, MessageBoxIcon.Information)
                    End If
                End If
            End If
        Else
            MessageBox.Show("Unesite valjan broj.", "Greška",
                MessageBoxButtons.OK, MessageBoxIcon.Warning)
        End If
    End Sub
    Private Function JeProstBroj(ByVal broj As Integer) As Boolean
        If broj <= 1 Then
            Return False
        End If

        For i As Integer = 2 To Math.Sqrt(broj)
            If broj Mod i = 0 Then
                Return False
            End If
        Next
        Return True
    End Function
End Class
```

2.2.20. Pretvarač mjernih jedinica

Ovaj program omogućava korisniku da unese broj i odabere odgovarajuću mjernu jedinicu (npr. centimetri, milimetri, kilometri itd.). Nakon unosa broja i odabira jedinice, program vrši pretvaranje unesenog broja u metre prema odabranoj jedini i prikazuje rezultat pretvaranja. Na primjer, ako unesemo broj u centimetrima, program će ga pretvoriti u metre i prikazati rezultat u metrima. Ovo omogućava brzu i jednostavnu konverziju različitih mjernih jedinica u metrima.



Slika 34 - Prikaz forme u VB Studio zadatak Br.20

```
Public Class Form1
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        cboJedinice.Items.Add("Centimetri (cm)")
        cboJedinice.Items.Add("Milimetri (mm)")
        cboJedinice.Items.Add("Kilometri (km)")
        cboJedinice.Items.Add("Milje (mi)")
        cboJedinice.Items.Add("Stope (ft)")
        cboJedinice.Items.Add("Inči (in)")
        cboJedinice.SelectedIndex = 0
    End Sub
    Private Sub btnPretvori_Click(sender As Object, e As EventArgs) Handles
        btnPretvori.Click
            Dim jedinica As Decimal
            Dim broj As Decimal
```

```

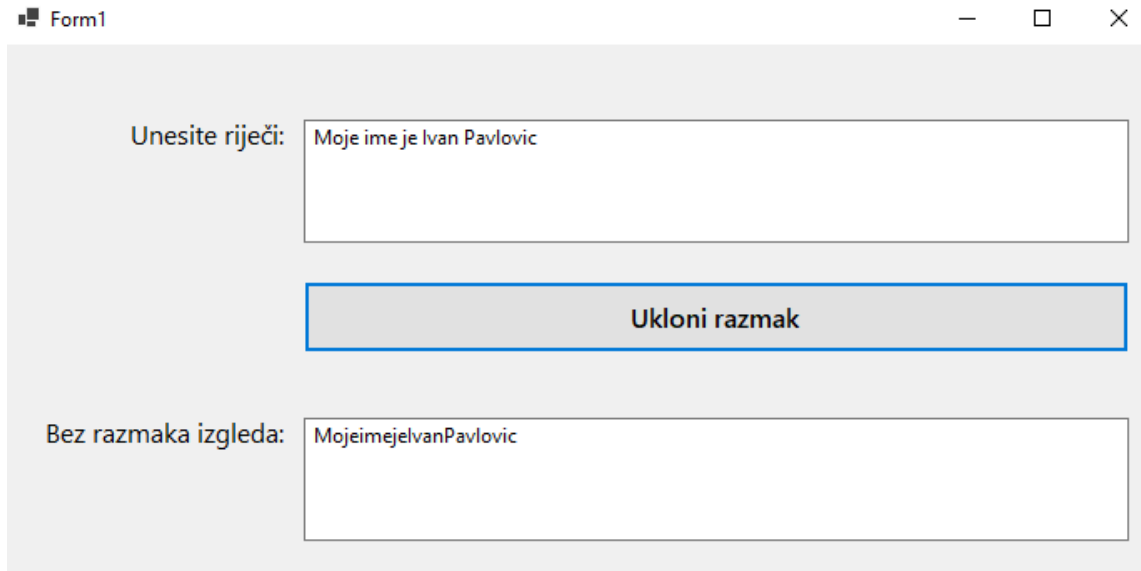
If Decimal.TryParse(txtBroj.Text, broj) Then
    If cboJedinice.SelectedIndex >= 0 Then
        Dim faktor As Decimal
        Select Case cboJedinice.SelectedItem.ToString()
            Case "Centimetri (cm)"
                faktor = 0.01
            Case "Milimetri (mm)"
                faktor = 0.001
            Case "Kilometri (km)"
                faktor = 1000
            Case "Milje (mi)"
                faktor = 1609.34
            Case "Stope (ft)"
                faktor = 0.3048
            Case "Inči (in)"
                faktor = 0.0254
        End Select
        Dim rezultat As Decimal = broj * faktor
        lblRezultat.Text = $"Rezultat: {broj} {cboJedinice.SelectedItem} = {rezultat}m"
    Else
        MessageBox.Show("Odaberite mernu jedinicu.", "Greška",
            MessageBoxButtons.OK, MessageBoxIcon.Error)
    End If
Else
    MessageBox.Show("Unesite valjani broj.", "Greška",
        MessageBoxButtons.OK, MessageBoxIcon.Error)
End If
End Sub
Private Sub btnNoviUnos_Click(sender As Object, e As EventArgs) Handles
    btnNoviUnos.Click
        txtBroj.Text = ""
        lblRezultat.Text = "Rezultat: "
    End Sub
End Class

```

2.2.21. Rečenica bez razmaka

Ovaj program omogućava korisniku unos rečenice sa razmacima. Nakon unosa rečenice, korisnik može kliknuti na dugme „Ukloni razmak“, nakon čega će program ukloniti sve razmake iz te rečenice. Rezultat, rečenica bez razmaka, prikazuje se u drugom tekstualnom polju.

Ovaj program može biti koristan za obradu i pripremu teksta, na primjer, za analizu ili obradu podataka.



Slika 35 - Prikaz forme u VB Studio zadatak Br.21

```
Public Class Form1
    Private Sub btnIzvrši_Click(sender As Object, e As EventArgs) Handles
        btnIzvrši.Click
            Dim rijeci As String
            rijeci = txtBoxRecenica.Text

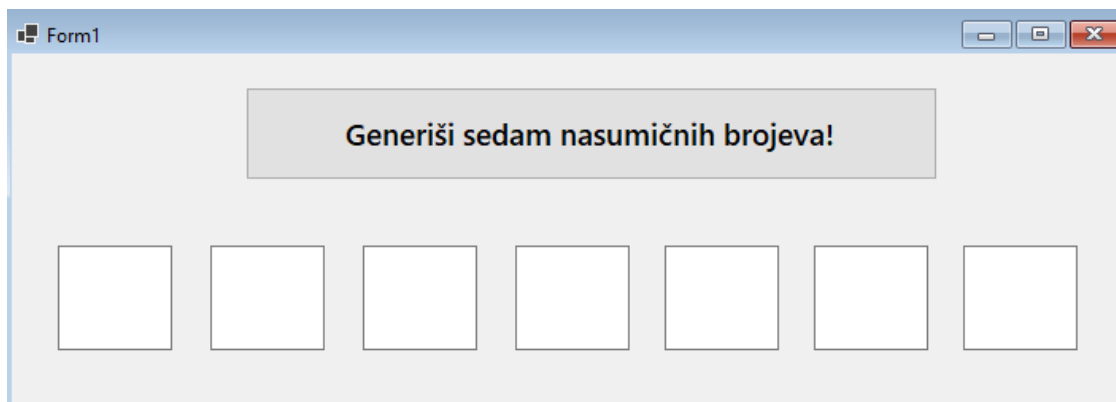
            rijeci = rijeci.Replace(" ", "")

            txtBoxBezRazmaka.Text = rijeci
        End Sub
    End Class
```

2.2.22. Generator nasumičnih brojeva od 1 do 50

Ovaj program služi za generisanje i prikazivanje sedam nasumično odabranih brojeva u opsegu od 1 do 50. Korisnik može kliknuti na dugme „Generiši sedam nasumičnih brojeva!“, nakon čega će program odabrati sedam brojeva nasumično i prikazati ih u odgovarajućim poljima za unos teksta.

Ovaj program može biti koristan, na primjer, za generisanje brojeva za lutriju ili bilo koje druge igre sa nasumičnim brojevima.



Slika 36 Prikaz forme u VB Studio zadatak Br.22

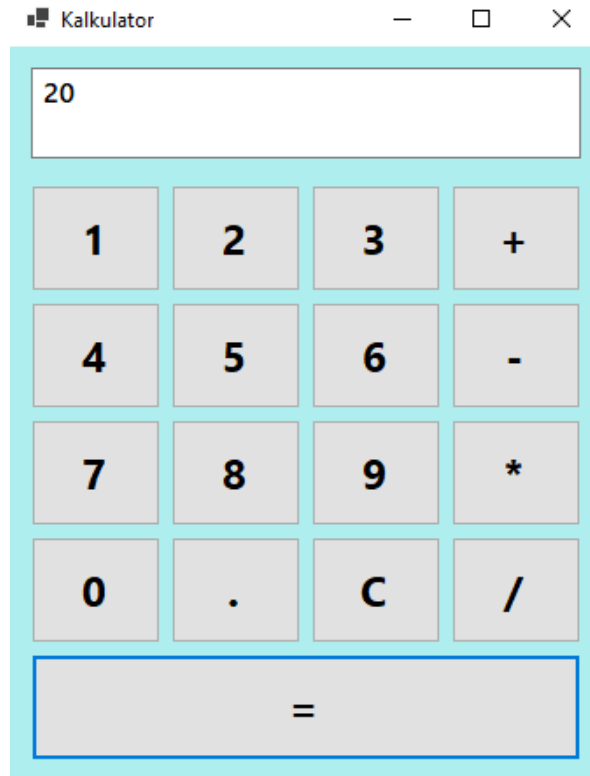
```
Public Class Form1
    Private Sub btnGeneriraj_Click(sender As Object, e As EventArgs)
        Dim r As New Random()
        Dim brojevi(6) As Integer

        For i As Integer = 0 To 6
            brojevi(i) = r.Next(1, 51)
        Next
        txtLoptica1.Text = brojevi(0).ToString()
        txtLoptica2.Text = brojevi(1).ToString()
        txtLoptica3.Text = brojevi(2).ToString()
        txtLoptica4.Text = brojevi(3).ToString()
        txtLoptica5.Text = brojevi(4).ToString()
        txtLoptica6.Text = brojevi(5).ToString()
        txtLoptica7.Text = brojevi(6).ToString()
    End Sub

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    End Sub
End Class
```

2.2.23. Kalkulator

Ovaj program prikazuje pojednostavljen kalkulator koji omogućava unos brojeva i izvođenje osnovnih matematičkih operacija kao što su sabiranje, oduzimanje, množenje i dijeljenje s dva broja. Također, omogućava se unos decimalnih brojeva i ima funkciju za brisanje unosa. Korisnik može unositi brojeve pritiskom na odgovarajuća dugmad za brojeve i matematičke operacije, a rezultat se prikazuje u TextBox polju. Program pravilno rukuje s različitim unosima i prikazuje rezultate na osnovu izabrane operacije.



Slika 37 - Prikaz forme u VB Studio zadatak Br.23

```
Public Class Form1
    Dim prviBroj As Decimal
    Dim drugiBroj As Decimal
    Dim operacija As Integer
    Dim znak As Boolean = False

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        UnesiBroj(1)
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        UnesiBroj(2)
    End Sub

    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        UnesiBroj(3)
    End Sub

    Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
        UnesiBroj(4)
    End Sub

    Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
        UnesiBroj(5)
    End Sub

    Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
```

```

    UnesiBroj(6)
End Sub

Private Sub Button7_Click(sender As Object, e As EventArgs) Handles Button7.Click
    UnesiBroj(7)
End Sub

Private Sub Button8_Click(sender As Object, e As EventArgs) Handles Button8.Click
    UnesiBroj(8)
End Sub

Private Sub Button9_Click(sender As Object, e As EventArgs) Handles Button9.Click
    UnesiBroj(9)
End Sub

Private Sub btnNula_Click(sender As Object, e As EventArgs) Handles btnNula.Click
    UnesiBroj(0)
End Sub

Private Sub btnObrisi_Click(sender As Object, e As EventArgs) Handles
btnObrisi.Click
    txtUnos.Text = "0"
End Sub

Private Sub btnTacka_Click(sender As Object, e As EventArgs) Handles
btnTacka.Click
    If Not (txtUnos.Text.Contains(".")) Then
        txtUnos.Text += "."
    End If
End Sub

Private Sub btnPlus_Click(sender As Object, e As EventArgs) Handles btnPlus.Click
    PrvaOperacija(1)
End Sub

Private Sub btnMinus_Click(sender As Object, e As EventArgs) Handles
btnMinus.Click
    PrvaOperacija(2)
End Sub

Private Sub btnMnozenje_Click(sender As Object, e As EventArgs) Handles
btnMnozenje.Click
    PrvaOperacija(3)
End Sub

Private Sub btnDjeljenje_Click(sender As Object, e As EventArgs) Handles
btnDjeljenje.Click
    PrvaOperacija(4)
End Sub

Private Sub btnJednako_Click(sender As Object, e As EventArgs) Handles
btnJednako.Click
    DrugaOperacija()
End Sub

Private Sub UnesiBroj(broj As Integer)
    If txtUnos.Text <> "0" Then
        txtUnos.Text += broj.ToString()
    End If
End Sub

```

```

Else
    txtUnos.Text = broj.ToString()
End If
End Sub

Private Sub PrvaOperacija(oper As Integer)
    prviBroj = Decimal.Parse(txtUnos.Text)
    txtUnos.Text = "0"
    znak = True
    operacija = oper
End Sub

Private Sub DrugaOperacija()
    If znak = True Then
        drugiBroj = Decimal.Parse(txtUnos.Text)
        Select Case operacija
            Case 1
                txtUnos.Text = (prviBroj + drugiBroj).ToString()
            Case 2
                txtUnos.Text = (prviBroj - drugiBroj).ToString()
            Case 3
                txtUnos.Text = (prviBroj * drugiBroj).ToString()
            Case 4
                If drugiBroj = 0 Then
                    txtUnos.Text = "Greška!"
                Else
                    txtUnos.Text = (prviBroj / drugiBroj).ToString()
                End If
            End Select
        znak = False
    End If
End Sub
End Class

```

2.2.24. Kalkulator razlike kvadrata

Ovaj program je jednostavan kalkulator koji omogućava korisnicima da unesu dva broja i odaberu između dve različite matematičke operacije. Korisnik može da unese brojeve u odgovarajuća polja za unos, odabrati željenu operaciju putem radio dugmati, a zatim pritisnuti dugme za izračunavanje. Rezultat selektovane operacije će biti prikazan na ekranu.

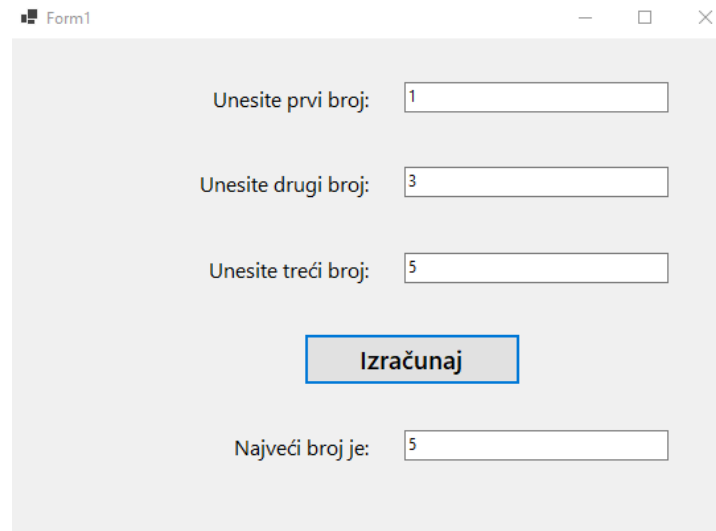
Također, program ima opciju za resetovanje unosa i zatvaranje aplikacije. Ovo je koristan alat za brzo izračunavanje kvadrata razlike ili razlike kvadrata između dva broja.

Slika 38 - Prikaz forme u VB Studio zadatak Br.24

```
Public Class Form1
    Private Sub btnIzracunaj_Click(sender As Object, e As EventArgs) Handles
btnIzracunaj.Click
        Dim a As Double
        Dim b As Double
        Dim rez As Double
        If Double.TryParse(txtPrviBroj.Text, a) AndAlso
Double.TryParse(txtDrugiBroj.Text, b) Then
            If radioButtonRazlikaKvadrata.Checked Then
                rez = a ^ 2 - b ^ 2
                lblSkritten.Text = "Razlika kvadrata je: " & rez.ToString()
            Else
                rez = (a - b) ^ 2
                lblSkritten.Text = "Kvadrat razlike je: " & rez.ToString()
            End If
        Else
            lblSkritten.Text = "Nekorektan unos brojeva."
        End If
        lblSkritten.Visible = True
    End Sub
    Private Sub btnNoviUnos_Click(sender As Object, e As EventArgs) Handles
btnNoviUnos.Click
        txtPrviBroj.Text = ""
        txtDrugiBroj.Text = ""
        lblSkritten.Text = ""
        lblSkritten.Visible = False
    End Sub
    Private Sub btnIzlaz_Click(sender As Object, e As EventArgs) Handles
btnIzlaz.Click
        Me.Close()
    End Sub
End Class
```

2.2.25. Najveći broj

Ovaj program omogućava korisniku da unese tri broja i zatim prikazuje najveći od njih nakon što korisnik klikne na dugme „Izračunaj“. Ako neko od polja za unos brojeva ostane prazno, program prikazuje poruku o grešci.



Slika 39 - Prikaz forme u VB Studio zadatak Br.25

```
Public Class Form1
    Private Sub btnIzracunaj_Click(sender As Object, e As EventArgs) Handles
btnIzracunaj.Click
        If String.IsNullOrEmpty(txtPrvi.Text) OrElse
String.IsNullOrEmpty(txtDrugi.Text) OrElse
String.IsNullOrEmpty(txtTreci.Text) Then
            MessageBox.Show("Molimo vas unesite sva tri broja.", "Greška",
MessageBoxButtons.OK, MessageBoxIcon.Error)
            Return
        End If
        Dim a As Integer = Integer.Parse(txtPrvi.Text)
        Dim b As Integer = Integer.Parse(txtDrugi.Text)
        Dim c As Integer = Integer.Parse(txtTreci.Text)
        Dim najveći As Integer
        If a >= b AndAlso a >= c Then
            najveći = a
        ElseIf b >= a AndAlso b >= c Then
            najveći = b
        Else
            najveći = c
        End If
        txtNajveci.Text = najveći.ToString()
    End Sub
End Class
```

ZAKLJUČAK

Zaključak ovog seminarskog rada naglašava važnost Visual Basic.NET programskog jezika u razvoju aplikacija za platformu Windows. Kroz upotrebu objektno orijentisanog programiranje (OOP) i RAD (Rapid Application Development) alata, programeri mogu brzo i efikasno razvijati aplikacije koje poboljšavaju produktivnost i olakšavaju svakodnevni život korisnicima.

VB.NET se ističe kao moćan alat za razvoj modularnih i održivih aplikacija, zahvaljujući svojoj podršci za OOP principe. Ovaj jezik omogućava programerima da organizuju svoj kod u objekte sa definisanim svojstvima i metodama, čime olakšava upravljanje i održavanje aplikacija.

Kroz analizu razlika između konzolnih aplikacija i aplikacija sa grafičkim korisničkim interfejsom, razumijemo kako VB.NET može biti prilagodljiv za različite vrste projekata. U zaključku, VB.NET je važan jezik u svijetu softvera, a razvoj aplikacija u ovom jeziku donosi brojne prednosti. Kroz pravilno korištenje OOP principa i RAD alata, programeri mogu stvoriti funkcionalne i efikasne aplikacije koje doprinose unapređenju korisničkog iskustva i produktivnosti.

POPIS SLIKA

Slika 1 - Visual Basic.NET logo i logotip	2
Slika 2 - Provjera parnosti i neparnosti broja u VB konzoli	6
Slika 3 - Obrtanje unešenog stringa u konzoli	7
Slika 4 - Rezultat zadatka Br.3	9
Slika 5 - Rezultat zadatka Br.3 sa Stringom	9
Slika 6 - Rezultat zadatka Br.4	11
Slika 7 - Rezultat zadatka Br.5	13
Slika 8 - Prikaz forme u VB Studio Zadatak Br.1	14
Slika 9 - Rezultat BMI kalkulatora	16
Slika 10 - Korekcija greške u BMI kalkulatoru	16
Slika 11 - Prikaz forme u VB Studio Zadatak Br.2	17
Slika 12 - Rezultat Digitalnog Sata.....	18
Slika 13 - Prikaz forme u VB Studio Zadatak Br.3	19
Slika 14 - Rezultat Konvertera Valuta	21
Slika 15 - Prikaz forme u VB Studio Zadatak Br.4	22
Slika 16 - Rezultat analize teksta	23
Slika 17 - Prikaz forme u VB Studio Zadatak Br.5	24
Slika 18 - Rezultat Fibonačijevog niza	25
Slika 19 - Prikaz forme u VB Studio zadatak Br.6	26
Slika 20 - Rezultat Čitača Tekstualnih Datoteka	27
Slika 21 - Prikaz forme u VB Studio zadatak Br.7	28
Slika 22 - Prikaz forme u VB Studio zadatak Br.8	30
Slika 23- Prikaz forme u VB Studio zadatak Br.9	31
Slika 24 - Prikaz forme u VB Studio zadatak Br.10	32
Slika 25- Prikaz forme u VB Studio zadatak Br.11	34
Slika 26 - Prikaz forme u VB Studio zadatak Br.12	35
Slika 27 - Prikaz forme u VB Studio zadatak Br.13	36
Slika 28 - Prikaz forme u VB Studio zadatak Br.14	37
Slika 29 - Prikaz forme u VB Studio zadatak Br.15	38
Slika 30 - Prikaz forme u VB Studio zadatak Br.16	40
Slika 31 - Prikaz forme u VB Studio zadatak Br.17	41
Slika 32 - Prikaz forme u VB Studio zadatak Br.18	42
Slika 33 - Prikaz forme u VB Studio zadatak Br.19	43
Slika 34 - Prikaz forme u VB Studio zadatak Br.20	44
Slika 35 - Prikaz forme u VB Studio zadatak Br.21	46
Slika 36 - Prikaz forme u VB Studio zadatak Br.22	47
Slika 37 - Prikaz forme u VB Studio zadatak Br.23	48
Slika 38 - Prikaz forme u VB Studio zadatak Br.24	51
Slika 39 - Prikaz forme u VB Studio zadatak Br.25	52

CITATNI IZVORI

- [1] »Wikipedia,« [Mrežno]. Available: [https://en.wikipedia.org/wiki/Visual_Basic_\(.NET\)](https://en.wikipedia.org/wiki/Visual_Basic_(.NET)).
- [2] »Wikipedia,« [Mrežno]. Available:
https://en.wikipedia.org/wiki/List_of_.NET_libraries_and_frameworks.