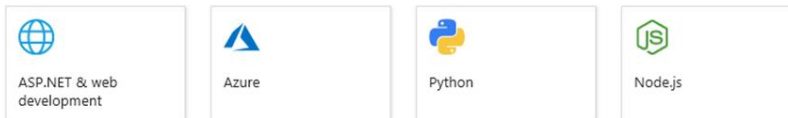


Prof. dr Zoran S. Veličković, dipl. inž. el.

.NET TEHNOLOGIJE

praktikum laboratorijskih vežbi

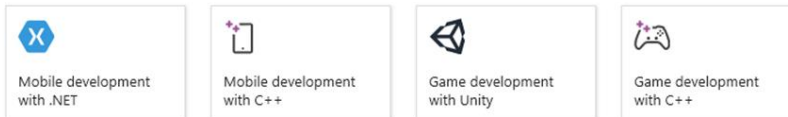
Web & cloud



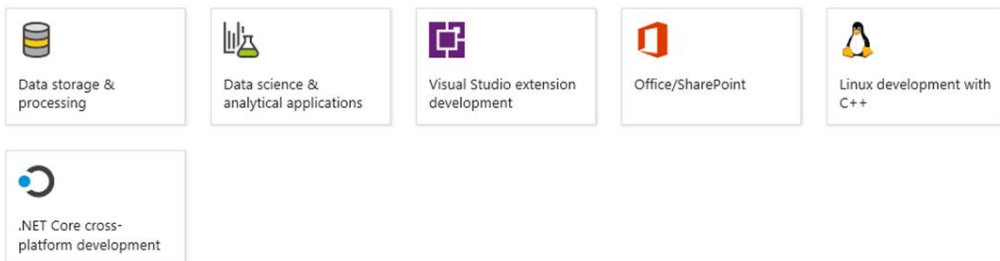
Windows



Mobile & gaming



Other toolsets



Niš, novembar 2021.

Student:

Broj indeksa:

Broj poena:

Potvrđuje:

NAPOMENE

LABORATORIJSKA VEŽBA 8: PREKLAPANJE OPERATORA

Cilj vežbe: Upoznavanje sa osnovnim pojmovima vezanim za preklapanja operatora. Preklapanje pojedinih operatora za operacije koje nisu direktno podržane u .NET-u.

Zadatak 1: Preklapanje aritmetičkog operatora sabiranja kod kompleksnih brojeva

Deklarisati klasu kompleksih brojeva `ComplexNumber` sa svojstvima `Real` i `Imaginary` tipa `double`. Sledeći aritmetičkih pravila za sabiranje kompleksnih brojeva, preklopiti operator sabiranja (+). Vrednosti realnih i imaginarnih delova kompleksnih brojeva učitati sa tastature primenom metode `Console.ReadLine()`. Formatirano štampati dobijene rezultate sabiranja unetih kompleksnih brojeva preklapanjem metode `ToString()`. Proveriti dobijene rezultate.

NAPOMENA: Obaviti konverziju tipa string dobijenog sa tastature u odgovarajući tip numerika.

Na slici 8-1 prikazan je izvorni kod klase `ComplexNumber` sa zadatim karakteristikama i programskim kodom za preklapanje operatora +. Na na slici 8-2 prikazan je izvorni kod glavnog programa koji primenjuje preklopljeni operator na objekte `ComplexNumber`.

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Vezba_8_Z1
{
    class ComplexNumber
    {
        public double Real
        {
            get;
            private set;
        }

        public double Imaginary
        {
            get;
            private set;
        }

        public ComplexNumber(double a, double b)
        {
            this.Real = a;
            this.Imaginary = b;
        }

        public override string ToString()
        {
            return string.Format("{0} {1} {2}i", this.Real, (Imaginary < 0 ? "-" : "+"),
Math.Abs(this.Imaginary));
        }

        public static ComplexNumber operator + (ComplexNumber x, ComplexNumber y)
        {
            return new ComplexNumber(x.Real + y.Real, x.Imaginary + y.Imaginary);
        }
    }
}
```

```

    }
}
}

```

Slika 8-1. Izvorni kod klase `ComplexNumber`.

```

namespace Vezba_8_Z1
{
    class Program
    {
        static void Main(string[] args)
        {

            ComplexNumber x, y;

            Console.WriteLine("Unesite realni deo kompleksnog broja X: ");
            double realniDeo = Convert.ToDouble(Console.ReadLine());

            Console.WriteLine("Unesite imaginarni deo kompleksnog broja X: ");
            double imaginarniDeo = Convert.ToDouble(Console.ReadLine());

            x = new ComplexNumber(realniDeo, imaginarniDeo);

            Console.WriteLine("Unesite realni deo kompleksnog broja Y: ");
            realniDeo = Convert.ToDouble(Console.ReadLine());

            Console.WriteLine("Unesite imaginarni deo kompleksnog broja Y: ");
            imaginarniDeo = Convert.ToDouble(Console.ReadLine());

            y = new ComplexNumber(realniDeo, imaginarniDeo);

            Console.WriteLine();

            Console.WriteLine("{0} + {1} = {2}", x, y, x + y);

        }
    }
}

```

Slika 8-2. Izvorni kod glavnog programa za testiranje preklopljenog operatora sabiranja.

Na slici 8-3 prikazan je izgled konzolnog prozora posle starovanja glavnog programa.

```

Microsoft Visual Studio Debug Console
Unesite realni deo kompleksnog broja X: 2
Unesite imaginarni deo kompleksnog broja X: 3
Unesite realni deo kompleksnog broja Y: 4
Unesite imaginarni deo kompleksnog broja Y: 5

(2 + 3i) + (4 + 5i) = 6 + 8i

C:\Users\Zoran\source\repos\Vezba_8_Z1\bin\Debug\netcoreapp3.1\Vezba_8_Z1.exe (p
rocess 13116) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->
Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

```

Slika 8-3. Izgled konzolnog prozora posle startovanja aplikacije.

Zadatak 2: Preklapanje osnovnih aritmetičkih operatora kod kompleksnih brojeva

Izmeniti prethodno napisani programski kod da bi se omogućilo kreiranje kompleksnih brojeva u glavnom programu. Potom, preklopiti aritmetičke operatore oduzimanja, množenja i deljenja kompleksnih brojeva. U glavnom programu obaviti testiranje svih preklopljenih operatora. Generišite kompleksne brojeve u skladu sa vašim brojem indeksa i prikazite dobijene rezultate.

Na slici 8-4 prikazan je kod glavnog programa, a na slici 8-5 kod preklopljenih operatora za realizaciju Zadatka 2.

```
namespace Vezba_8_Z2
{
    class Program
    {
        static void Main(string[] args)
        {
            {
                ComplexNumber x, y;
                x = new ComplexNumber(2.0, 2.0);
                y = new ComplexNumber(3.0, 3.0);

                Console.WriteLine();

                Console.WriteLine("{0} + {1} = {2}", x, y, x + y);
                Console.WriteLine("{0} * {1} = {2}", x, y, x * y);
                Console.WriteLine("{0} / {1} = {2}", x, y, x / y);
                Console.WriteLine("{0} - {1} = {2}", x, y, x - y);
            }
        }
    }
}
```

Slika 8-4. Izvorni kod glavnog programa za testiranje preklopljenih operatora.

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Vezba_8_Z2
{
    class ComplexNumber
    {
        public double Real
        {
            get;
            private set;
        }

        public double Imaginary
        {
            get;
            private set;
        }
    }
}
```

```

public ComplexNumber(double a, double b)
{
    this.Real = a;
    this.Imaginary = b;
}

public override string ToString()
{
    return string.Format("{0} {1} {2}i", this.Real, (Imaginary < 0 ? "-" : "+"),
Math.Abs(this.Imaginary));
}

public static ComplexNumber operator+ (ComplexNumber x, ComplexNumber y)
{
    return new ComplexNumber(x.Real + y.Real, x.Imaginary + y.Imaginary);
}

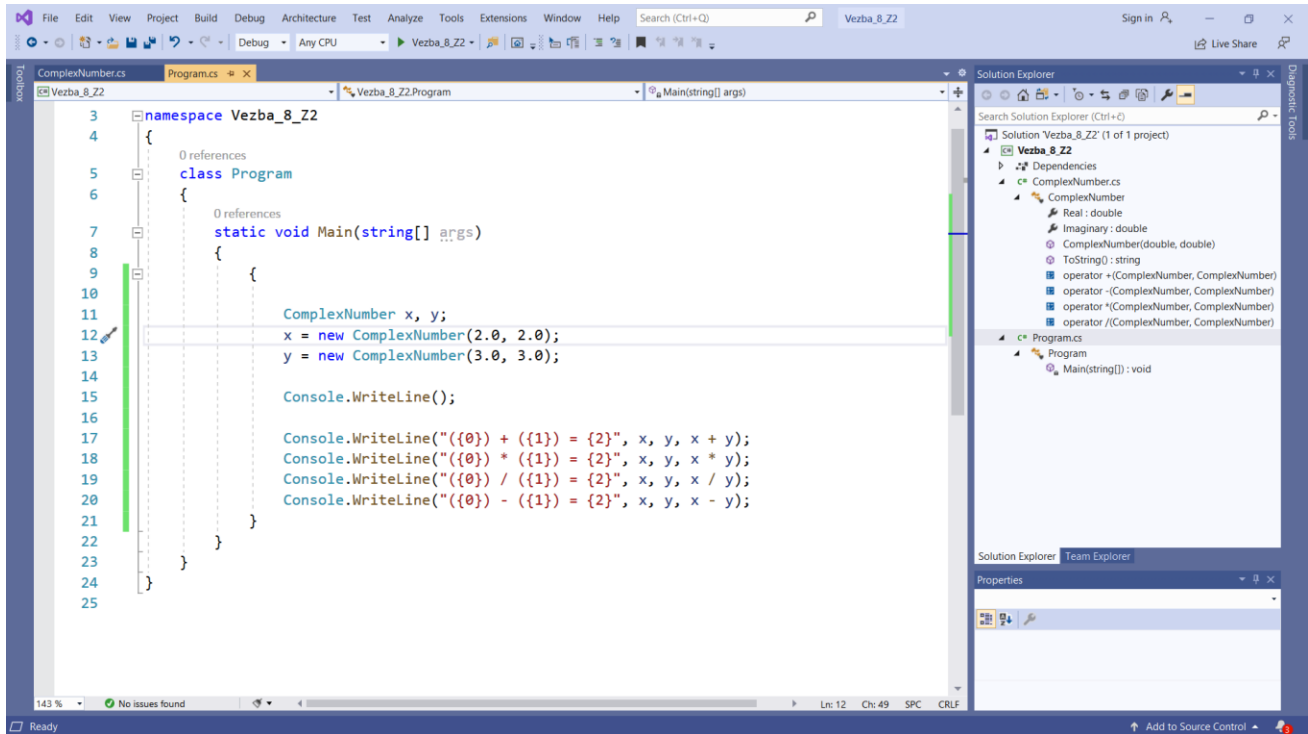
public static ComplexNumber operator- (ComplexNumber x, ComplexNumber y)
{
    return new ComplexNumber(x.Real - y.Real, x.Imaginary - y.Imaginary);
}

public static ComplexNumber operator* (ComplexNumber x, ComplexNumber y)
{
    return new ComplexNumber(x.Real * y.Real - x.Imaginary * y.Imaginary, x.Real *
y.Imaginary + x.Imaginary * y.Real);
}

public static ComplexNumber operator/ (ComplexNumber x, ComplexNumber y)
{
    double imenilac = y.Real * y.Real + y.Imaginary * y.Imaginary;
    return new ComplexNumber( (x.Real * y.Real + x.Imaginary *
y.Imaginary)/imenilac, (x.Imaginary * y.Real - x.Real * y.Imaginary)/imenilac);
}
}
}

```

Slika 8-5. Izvorni kod klase preklopljenih operatora.



Slika 8-6. Projekat programa za testiranje preklopljenih aritmetičkih operatora u VS-u. Na slici 8-6 prikazan je izgled projekata za Zadatak 2 realizovan u VS-u.

Ispisati sadržaj ekrana konzolnog ekrana za tražene kompleksne brojeve. Da li odgovaraju Vašim očekivanjima?



Zadatak 3: Preklapanje operatora poređenja kod kompleksnih brojeva

Na bazi prethodno napisanih programskih kodova preklopiti operator konjugovanja (\sim), kao i operatore poređenja jednakosti ($==$) i nejednakosti ($!=$). U glavnom programu kreirati tri kompleksna broja, napraviti razliku prvog i drugog konjugovanog broja, a potom proveriti jednakost dobijenog rezultata i trećeg kompleksnog broja. Štampati dobijene rezultate. U predviđenom prostoru dopisati delove klase `ComplexNumber` koji se odnose na tražene operatore.

Prostor za programski kod

Prostor za prikaz konzolnog prozora

Zadatak 4: Preklapanje operatora sabiranja i oduzimanja klase Kutija

Deklarisati klasu Kutija sa privatnim poljima Sirina, Duzina i Visina. Preklopiti operator sabiranja (+) i oduzimanja (-) tako što će povratni tip biti takođe Kutija sa odgovarajućim dimenzijama.

NAPOMENA: Polja dobijene Kutija se dobijaju sabiranjem, odnosno, oduzimanjem odgovarajućih polja tipova Kutija.

U glavnom programu kreirati tri tipa Kutija, napraviti zbir i razliku prvog i drugog objekta Kutija. Štampati dobijene rezultate. Da li dobijeni rezultati odgovaraju uslovima zadatka?

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Vezba_8_Z4
{
    class Box
    {
        private double length;    // dužina kutije
        private double breadth;   // širina kutije
        private double height;    // visina kutije

        public double getVolume()
        {
            return length * breadth * height;
        }
        public void setLength(double len)
        {
            this.length = len;
        }
        public void setBreadth(double bre)
        {
            this.breadth = bre;
        }
    }
}
```

```

    }
    public void setHeight(double hei)
    {
        this.height = hei;
    }

    // Preklapanje + operatora.
    public static Box operator +(Box b, Box c)
    {
        Box box = new Box();
        box.length = b.length + c.length;
        box.breadth = b.breadth + c.breadth;
        box.height = b.height + c.height;
        return box;
    }

    public static Box operator - (Box b, Box c)
    {
        Box box = new Box();
        box.length = b.length - c.length;
        box.breadth = b.breadth - c.breadth;
        box.height = b.height - c.height;
        return box;
    }
}
}
}

```

Slika 8-7. Izvorni kod preklopljenih operatora klase Box.

```

using System;

namespace Vezba_8_Z4
{
    class Program
    {
        static void Main(string[] args)
        {
            Box Box1 = new Box(); // Deklaracija Box1
            Box Box2 = new Box(); // Deklaracija Box2
            Box Box3 = new Box(); // Deklaracija Box3
            Box Box4 = new Box(); // Deklaracija Box4
            double volume = 0.0; // Polje za čuvanje zapremine kutije

            // Box 1
            Box1.setLength(6.0);
            Box1.setBreadth(7.0);
            Box1.setHeight(5.0);

            // Box 2
            Box2.setLength(12.0);
            Box2.setBreadth(13.0);
            Box2.setHeight(10.0);

            // Zapremina Box 1
            volume = Box1.getVolume();
            Console.WriteLine("Volume of Box1 : {0}", volume);
        }
    }
}

```

```

// Zapremina Box 2
volume = Box2.getVolume();
Console.WriteLine("Volume of Box2 : {0}", volume);

// Dodavanje i oduzimanje dva objekta tipa Box
Box3 = Box1 + Box2;
Box4 = Box1 - Box2;

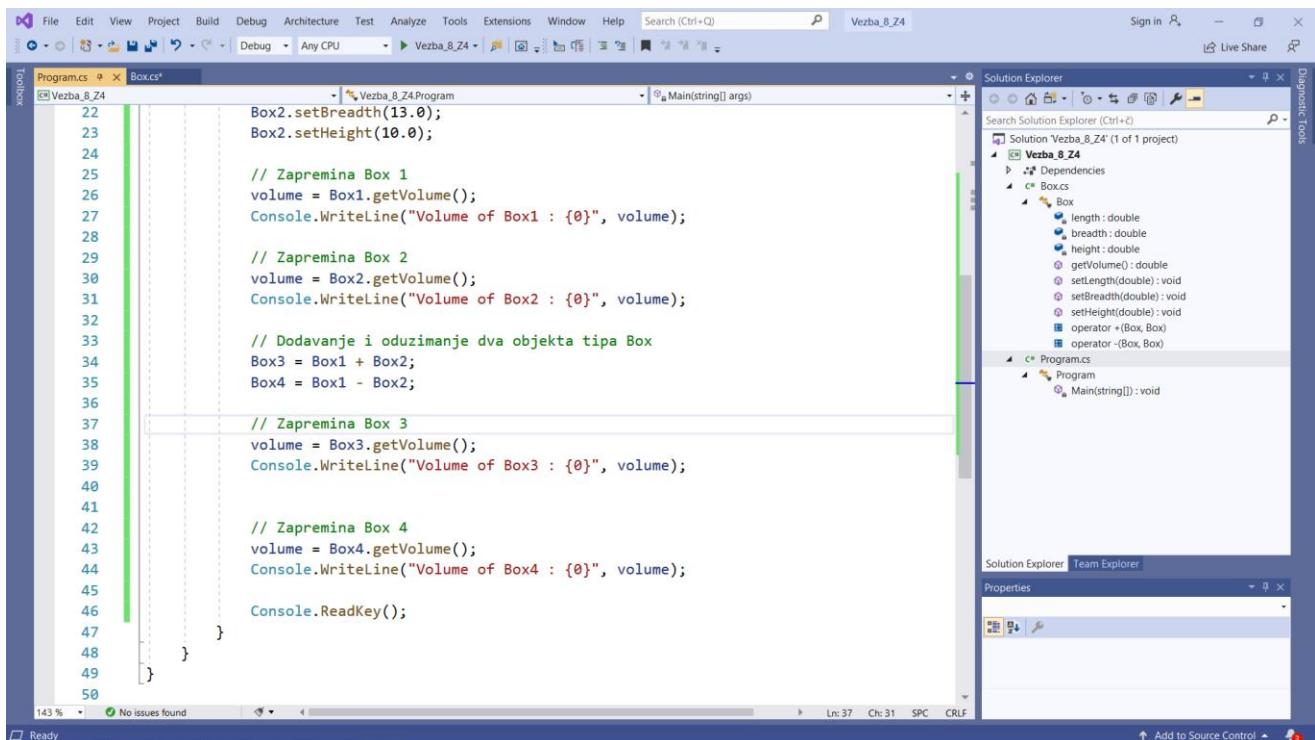
// Zapremina Box 3
volume = Box3.getVolume();
Console.WriteLine("Volume of Box3 : {0}", volume);

// Zapremina Box 4
volume = Box4.getVolume();
Console.WriteLine("Volume of Box4 : {0}", volume);

Console.ReadKey();
}
}
}

```

Slika 8-8. Izvorni kod glavnog programa za testiranje preklopljenih operatora klase Box.



Slika 8-9. Projekat programa za testiranje preklopljenih operatora klase Box u VS-u.

Zadaci za samostalan rad

1. U delu za testiranje kreirati dva kompleksna broja i testirati sve operacije. Proveriti da li su kreirani kompleksni brojevi jednaki. Kreirati nova dva kompleksna broja takva da posle konjugovanja prvog od njih dobijate drugi. Prikazati dobijene rezultate.

Prostor za prikaz konzolnog prozora

2. Kreirati četiri kompleksna broja koja u Re-Im ravni formiraju pravougaonik. Korišćenjem preklapljenih operatora proveriti da li se zaista kreira pravougaonik, a potom izračunati površinu tako dobijenog pravougaonika.
3. Kreirati niz kompleksnih brojeva. Potom, kreirati novi niz istih kompleksnih brojeva koji su predstavljeni u trigonometrijskom obliku.

NAPOMENA:

Trigonometrijski oblik kompleksnog broja se sastoji od argumenta (Argument) i ugla (Ugao). Za izračunavanje ugla trigonometrijskog broja koristiti `Math.Ath2(im, re)`, a Argument se određuje primenom Pitagorine teoreme nad katetama kompleksnog broja (`Math.sqrt(re*re + im*im)`).

Prostor za programski kod

U Nišu	POTVRĐUJE