

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА  
ПОЛІТЕХНІКА”

Кафедра систем автоматизованого проектування

**Звіт**

З дисципліни

«Екстремальне програмування»

Лабораторної роботи №6

«TDD. Калькулятор»

Варіант №14

Виконав:

Студент гр.  
КН-311

Куриляк Ю.А.

Прийняв:

Щербак С.С.

Львів – 2020

## Теоретичні відомості:

Керована тестами розробка (КТР), Розробка через тестування ([англ. \*Test-driven development \(TDD\)\*](#)) — [технологія розробки програмного забезпечення](#), яка використовує короткі ітерації розробки, що починаються з попереднього написання тестів, які визначають необхідні покращення або нові функції. Кожна ітерація має на меті розробити код, який пройде ці тести. Нарешті, програміст або група вдосконалюють код для погодження змін. Один із ключових моментів TDD полягає у тому, що підготовка тестів перед написанням самого коду пришвидшує процес внесення змін. Варто зауважити, що керована тестами розробка є методологією розробки програмного забезпечення, а не його [тестування](#).

Test-Driven Development відноситься до концепції [екстремального програмування](#), яка стверджує, що спершу потрібно писати тести, а вже потім код, яка веде свій початок з [1999](#) року, однак, останнім часом спостерігається загальніший інтерес до даної методології.

Програмісти також використовують дану методологію для вдосконалення і [зневадження сирцевого коду](#), раніше написаного з використанням інших методологій розробки.

## Виконання роботи:

calculator code

```
namespace XPCalculator
{
    public class Calculator
    {
        public Calculator()
        {

        }

        public double sum(double a, double b)
        {
            return 0;
        }

        public double sub(double a, double b)
        {
            return 0;
        }
    }
}
```

```

    }

    public double mul(double a, double b)
    {
        return 0;
    }

    public double div(double a, double b)
    {
        return 0;
    }
}
}

```

## Tests code

```

namespace Calculator.Tests
{
    [TestClass]
    public class CalculatorTests
    {
        XPCalculator.Calculator calculator;

        [TestInitialize]
        public void TestInitialize()
        {
            calculator = new XPCalculator.Calculator();
        }

        [TestMethod]
        public void Sum_1and5_and_2and7_returned4and2()
        {
            double a = 1.5;
            double b = 2.7;
            double exp = 4.2;

            double actual = calculator.sum(a, b);

            Assert.AreEqual(exp, actual, "Wrong summed");
        }

        [TestMethod]
        public void Sub_1and5_and_2and7_returned4and2()
        {
            double a = 1.5;
            double b = 2.7;
            double exp = -1.2;

            double actual = calculator.sub(a, b);

            Assert.AreEqual(exp, actual, "Wrong subtracted");
        }
    }
}

```

```

[TestMethod]
public void Mul_1and5_and_2and7_returned4and2()
{
    double a = 1.5;
    double b = 2.7;
    double exp = 4.05;

    double actual = calculator.mul(a, b);

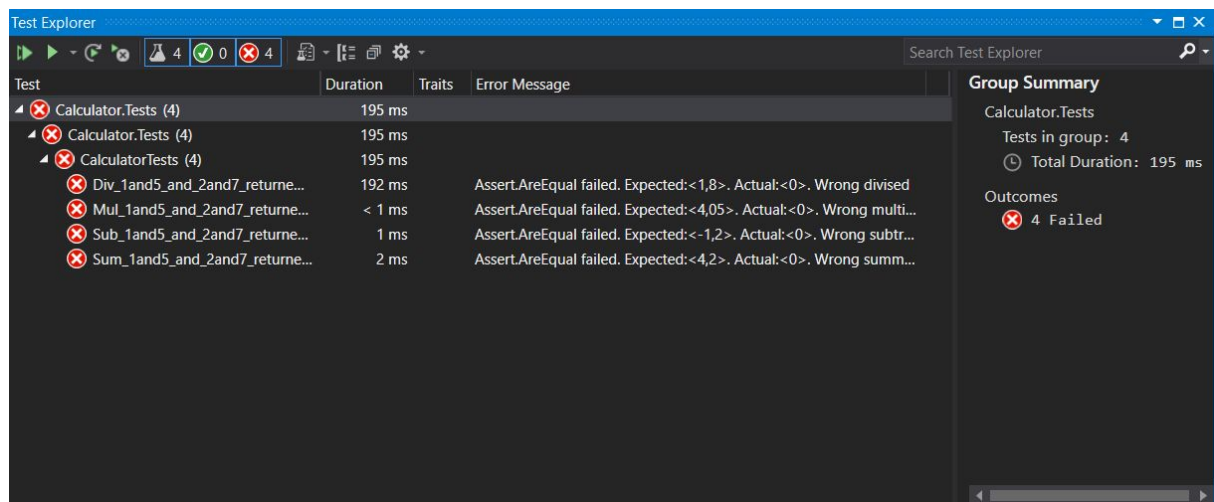
    Assert.AreEqual(exp, actual, "Wrong multiplied");
}

[TestMethod]
public void Div_1and5_and_2and7_returned4and2()
{
    double a = 1.5;
    double b = 2.7;
    double exp = 1.8;

    double actual = calculator.div(b,a);

    Assert.AreEqual(exp, actual, "Wrong divided");
}
}
}

```



Жоден з тестів не пройшов.

Написано наступний код для Calculator:

```

public class Calculator
{
    public Calculator()
    {
    }
}

```

```

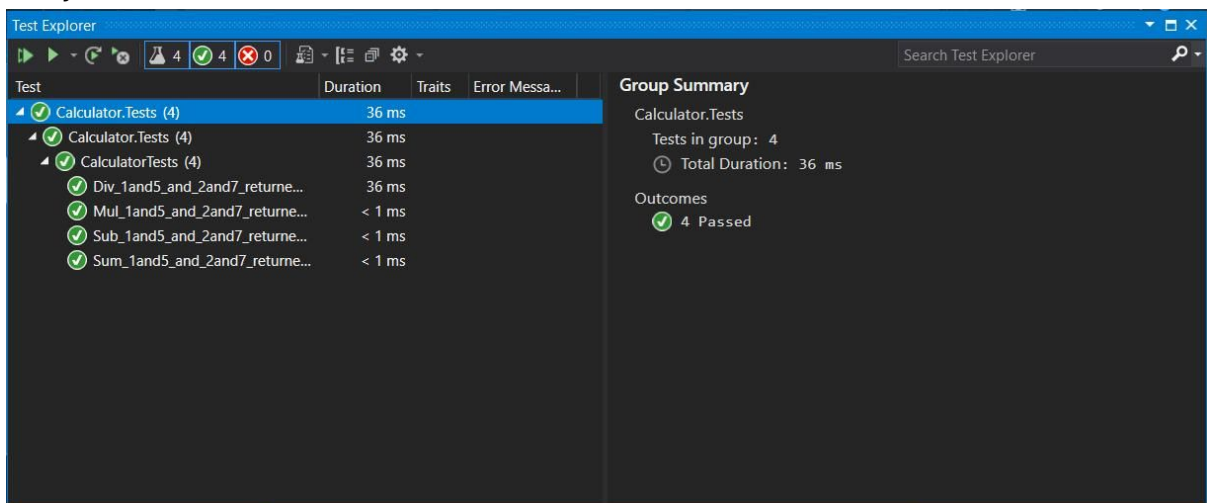
    public double sum(double a, double b)
    {
        return a + b;
    }

    public double sub(double a, double b)
    {
        return a - b;
    }

    public double mul(double a, double b)
    {
        return a * b;
    }

    public double div(double a, double b)
    {
        return a / b;
    }
}

```



Усі тести пройдено, отже, функції працюють як було задумано.

Напишемо фронтенд (C# wpf)

```

namespace XPCalculator
{
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            calculator = new Calculator();
            InitializeComponent();
        }
    }
}

```

```

int activeOperator = 0;
string consoleOutput;
double a = 0;
double b = 0;
double rez = 0;
Calculator calculator;

private void Button_Click_Sum(object sender, RoutedEventArgs e)
{
    activeOperator = 1;
    a = Convert.ToDouble(tb_console.Text.ToString());
    tb_console.Text = "";
}

private void Button_Click_Sub(object sender, RoutedEventArgs e)
{
    a = Convert.ToDouble(tb_console.Text.ToString());
    tb_console.Text = "";
    activeOperator = 2;
}

private void Button_Click_Mul(object sender, RoutedEventArgs e)
{
    a = Convert.ToDouble(tb_console.Text.ToString());
    tb_console.Text = "";
    activeOperator = 3;
}

private void Button_Click_Div(object sender, RoutedEventArgs e)
{
    a = Convert.ToDouble(tb_console.Text.ToString());
    tb_console.Text = "";
    activeOperator = 4;
}

private void Button_Click_Rez(object sender, RoutedEventArgs e)
{
    b = Convert.ToDouble(tb_console.Text.ToString());
    switch (activeOperator)
    {
        case 0:
            consoleOutput = "";
            break;

        case 1:
            consoleOutput = calculator.sum(a, b).ToString();
            break;

        case 2:
            consoleOutput = calculator.sub(a, b).ToString();
            break;
    }
}

```

```

        case 3:
            consoleOutput = calculator.mul(a, b).ToString();
            break;

        case 4:
            consoleOutput = calculator.div(a, b).ToString();
            break;
    }
    activeOperator = 0;
    tb_console.Text = consoleOutput;
}

private void Button_Click_Clear_Console(object sender, RoutedEventArgs e)
{
    activeOperator = 0;
    tb_console.Text = "";
}
}
}

```

## XAML:

```

<Window x:Class="XPCalculator.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:XPCalculator"
    mc:Ignorable="d"
    Title="MainWindow" Height="450" Width="800">
    <Grid Background="#6d6875">
        <TextBox Name="tb_console" Background="#ffcd2" FontSize="30"
Margin="100,100,100,270" />
        <Button Background="#b5838d" Content="+" FontSize="30"
HorizontalAlignment="Left" Margin="100,209,0,0" VerticalAlignment="Top" Width="75"
Click="Button_Click_Sum"/>
        <Button Background="#b5838d" Content="-" FontSize="30"
HorizontalAlignment="Left" Margin="277,209,0,0" VerticalAlignment="Top" Width="75"
Click="Button_Click_Sub"/>
        <Button Background="#b5838d" Content="*" FontSize="30"
HorizontalAlignment="Left" Margin="449,209,0,0" VerticalAlignment="Top" Width="75"
Click="Button_Click_Mul"/>
        <Button Background="#b5838d" Content="/" FontSize="30"
HorizontalAlignment="Left" Margin="619,209,0,0" VerticalAlignment="Top" Width="75"
Click="Button_Click_Div"/>
        <Button Background="#b5838d" Content="=" FontSize="30"
HorizontalAlignment="Left" Margin="722,100,0,0" VerticalAlignment="Top" Width="50"
Height="50" Click="Button_Click_Rez"/>
    </Grid>
</Window>

```

```
<Button Background="#b5838d" Content="Clear" FontSize="20"
HorizontalAlignment="Left" Margin="10,114,0,0" VerticalAlignment="Top" Width="75"
Click="Button_Click_Clear_Console"/>
</Grid>
</Window>
```

