

IVS - Projekt Kalkulačka

Tým xpetrr00

Autoři:

Radek Petr (xpetrr00)

Marek Fadrný (xfadrn02)

Duy Anh Pham Vu (xphamv00)

Oleksandr Tepliakov (xtepli02)



Radek Petr

- GUI, Testy, Makefile

Marek Fadrný

- Profiling, Instalátor

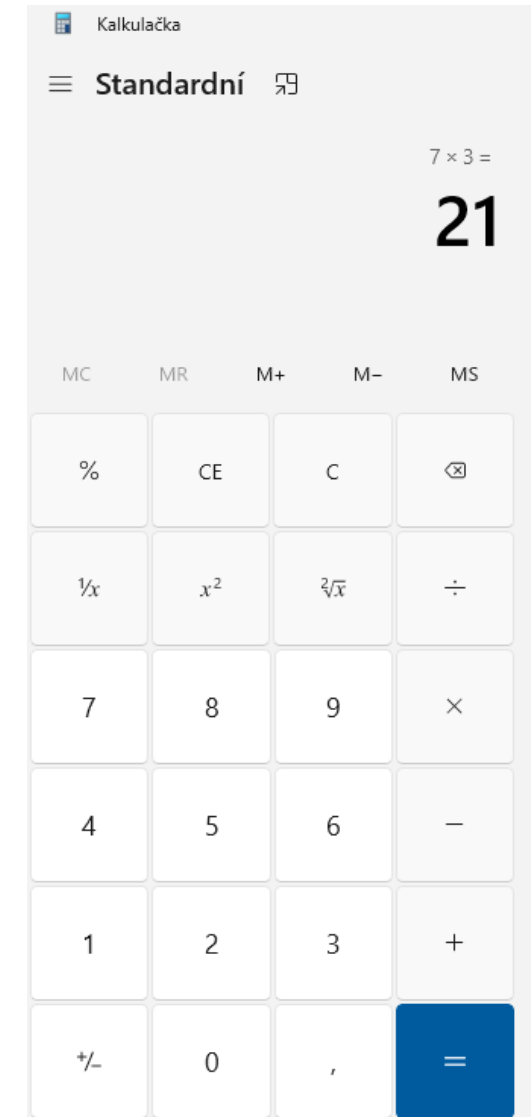
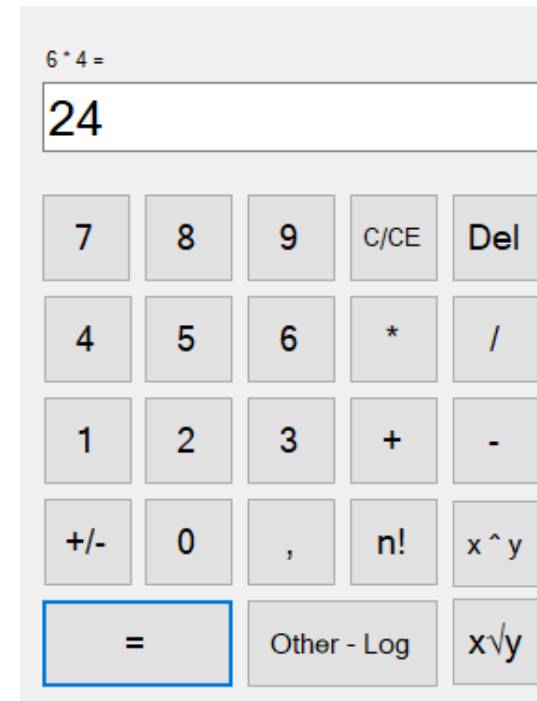
Duy Anh Pham Vu

- Základní matematické operace, Uživatelská příručka

Sasha Tepliakov

- Pokročilé matematické operace, Libovolná funkce

- Technologie - .NET
- GUI - WindowsForms
- Cílová platforma - Windows



```
public static double Add(double x, double y)
{
    if (x < 0 && y < 0)
    {
        if (Math.Max(x, y) < Double.MinValue + Math.Min(x, y))
        {
            throw new OverflowException();
        }
    }

    if (x > 0 && y > 0)
    {
        if (Math.Min(x, y) > Double.MaxValue - Math.Max(x, y))
        {
            throw new OverflowException();
        }
    }

    return x + y;
}
```

```
public static double GeneralRoot(double x, int n)
{
    if (x < 0 && n != 1 && n != -1)
    {
        throw new ArgumentException("Cannot calculate root of negative number");
    }
    else if (n == 0)
    {
        throw new ArgumentException("Cannot calculate 0-th root");
    }

    double result = Math.Pow(x, 1.0 / n);

    if (Double.IsInfinity(result))
    {
        throw new OverflowException();
    }

    return result;
}
```

```
public static double Divide(double x, double y)
{
    if (y == 0)
    {
        throw new DivideByZeroException();
    }

    if (x == 0)
    {
        return 0;
    }

    double result = x / y;

    if (Double.IsInfinity(result))
    {
        throw new OverflowException();
    }

    return result;
}
```

1 Úvod

Jednoduchá kalkulačka se základní funkcionalitou pro sčítání, odčítání, násobení a dělení čísel.

Dále podporuje výpočet faktoriálu, mocniny a odmocniny čísel.

Pro platformu Windows 64-bit.

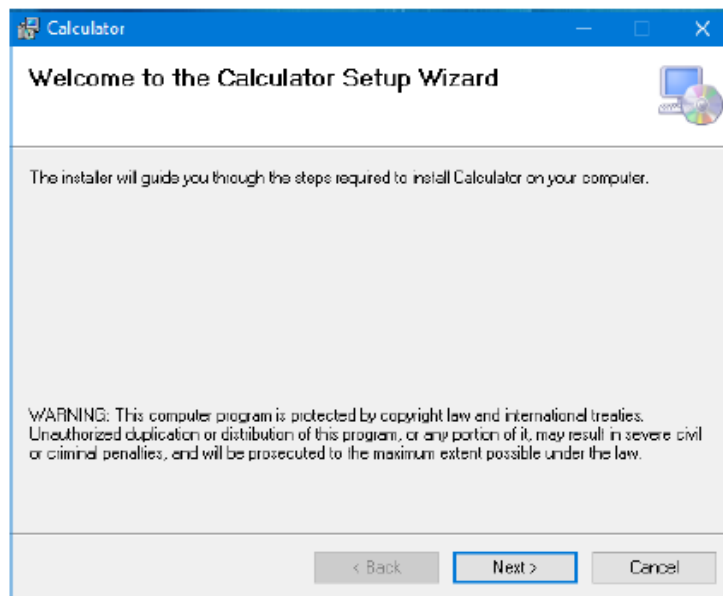
2 Instalace programu

Rozbalte .zip soubor na plochu

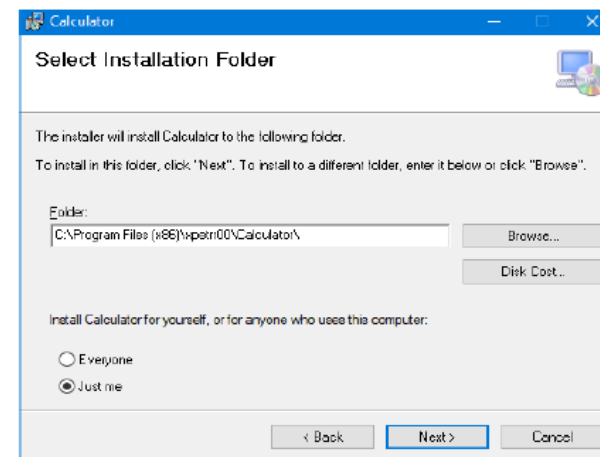
Na ploše se vytvoří 2 soubory: **CalculatorSetup.msi** a **setup.exe**

Spusťte jakýkoli jeden z nich

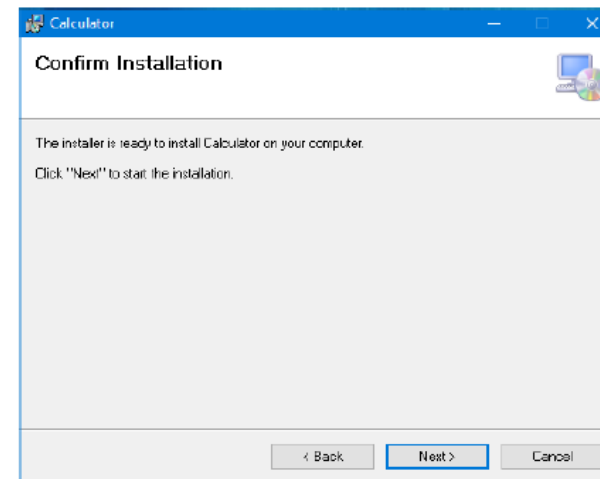
Otevře se nové instalační okno



Klikněte na **Next**

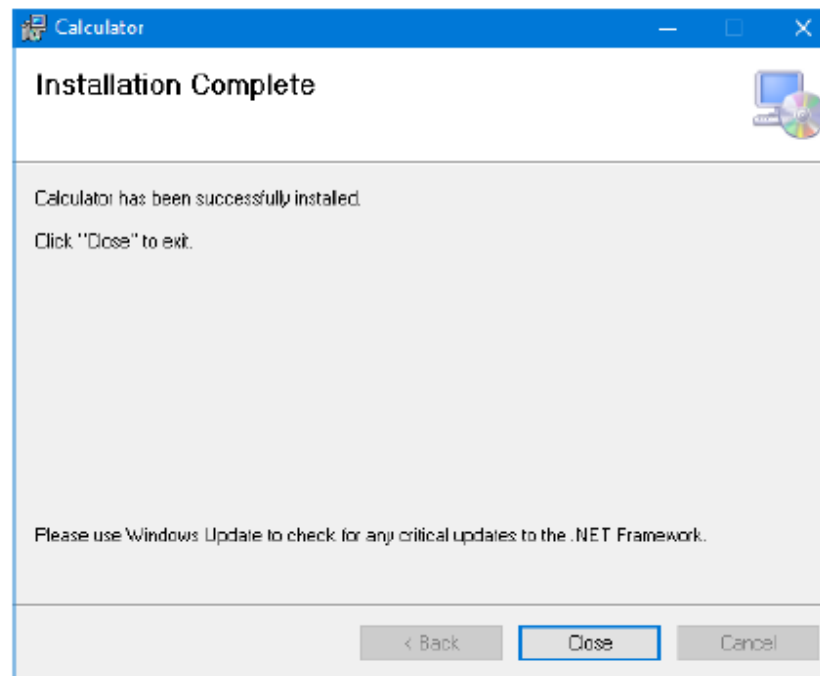


Klikněte na **Browse**, pokud chcete vybrat místo na disku, kam chcete program nainstalovat
Zvolte, zda chcete program nainstalovat pro všechny uživatele (**Everyone**) nebo pro sebe (**Just me**)
(Klikněte na **Disk Cost**, pokud chcete zjistit, kolik místa na disku bude program potřebovat)



Potvrďte instalaci programu kliknutím na **Next** a počkejte, než se program instaluje

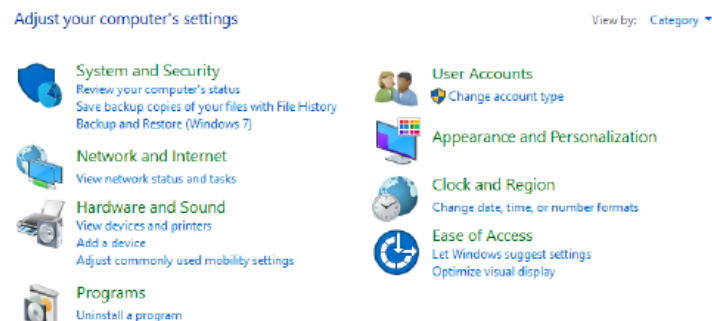
Vyskočí nové okno Windowsu pro oprávnění programu dělat změny na počítači, klikněte na **Yes**



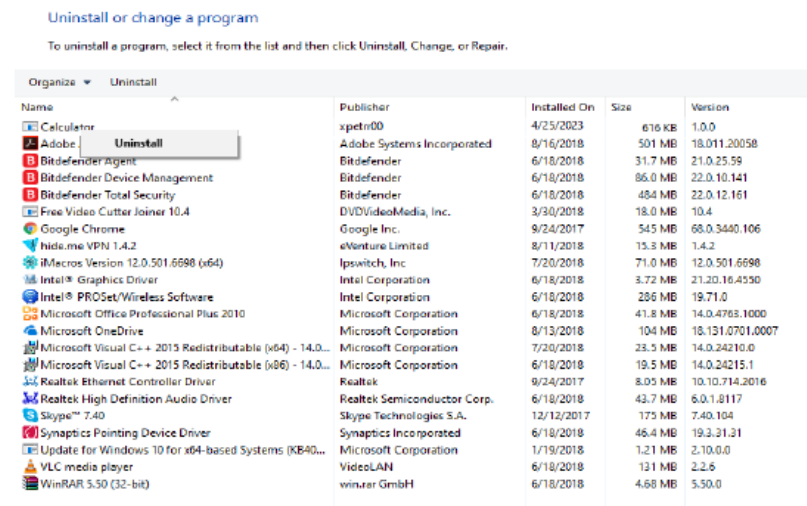
Po úspěšné instalaci zavřete okno instalátoru kliknutím na **Close**

3 Odinstalace programu

Spusťte **Windows Menu** a vyhledejte **Ovládací panel** (v angl. **Control Panel**)

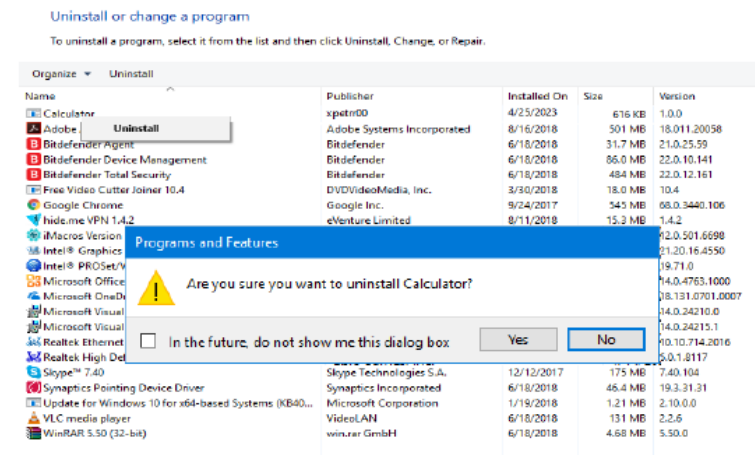


Klikněte na **Odinstalovat program** (Uninstall a program)



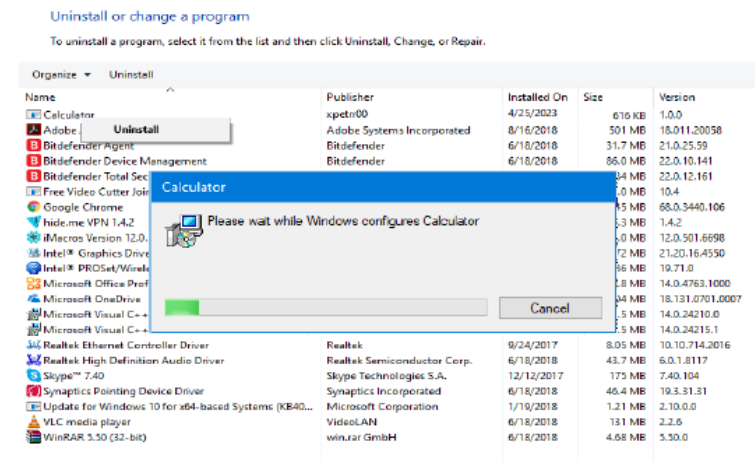
Klikněte pravým tlačítkem myši na **Calculator** a zvolte **Odinstalovat** (Uninstall)

Otevře se nové okno, potvrďte odinstalaci kliknutím na **Yes**



Vyskočí nové okno Windowsu pro oprávnění programu dělat změny na počítači, klikněte na **Yes**

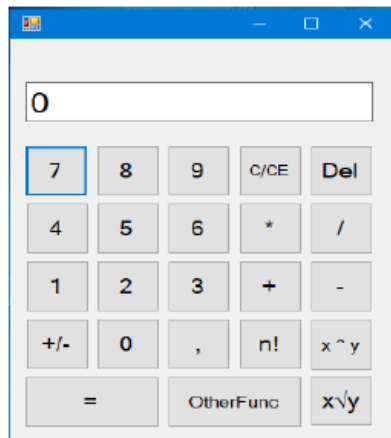
Počkejte, než instalátor odinstaluje program



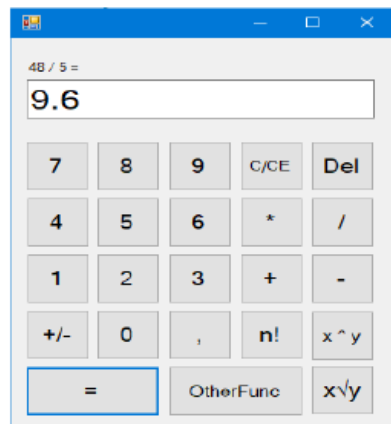
Až se program odinstaluje, instalátor se sám ukončí

4 Jak pracovat s programem

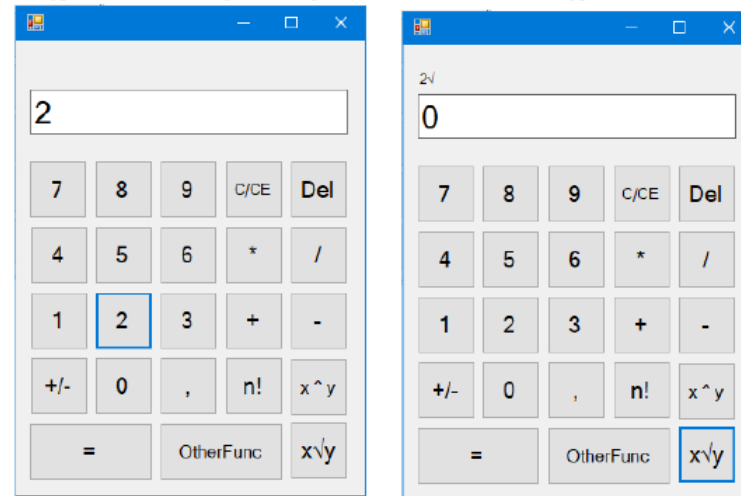
Spusťte kalkulačku pomocí zástupce **Calculator** na ploše



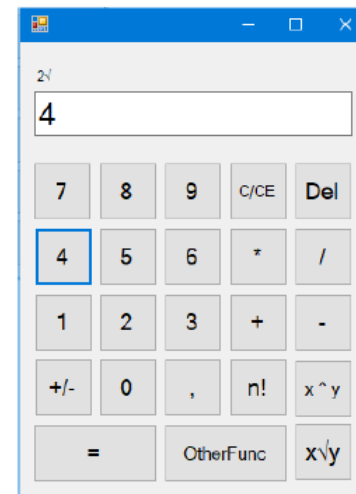
Zadejte výraz, který chcete vypočítat



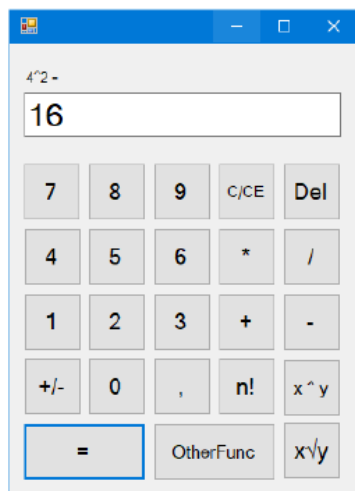
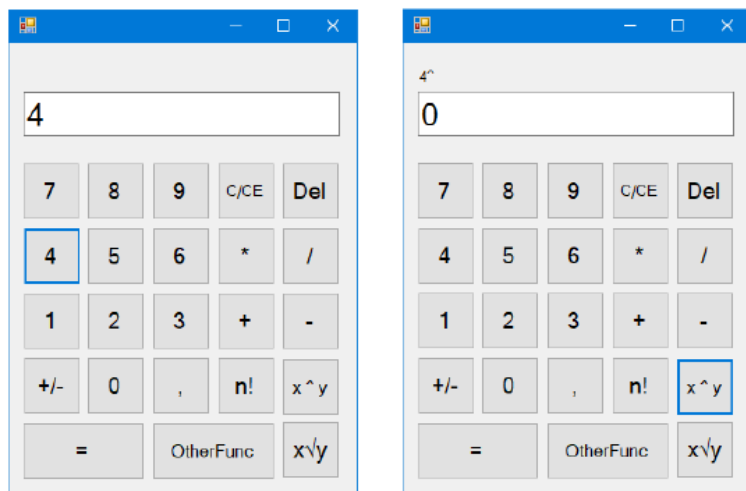
Pro výpočet n -té odmocniny čísla zadejte, kolikátou odmocninu chcete vypočítat



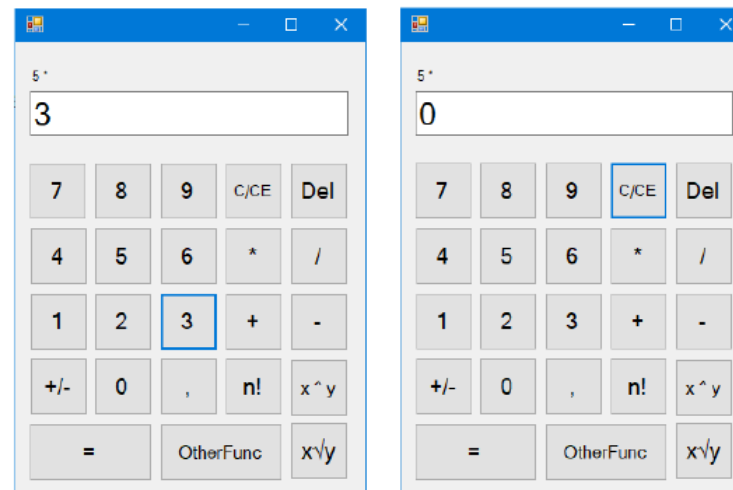
Poté stisknete tlačítko odmocniny a zadejte, odmocninu kterého čísla chcete vypočítat



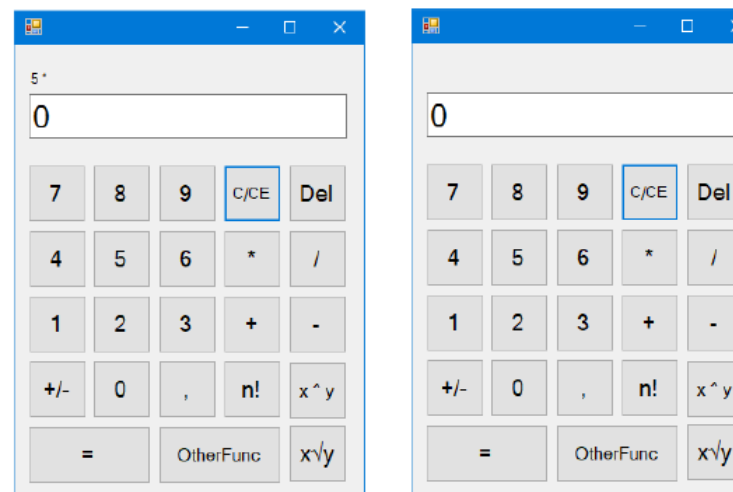
Pro výpočet n -té mocniny čísla nejdřív zadejte **základ mocniny**, poté klikněte na tlačítko mocniny a nakonec zadejte **exponent**



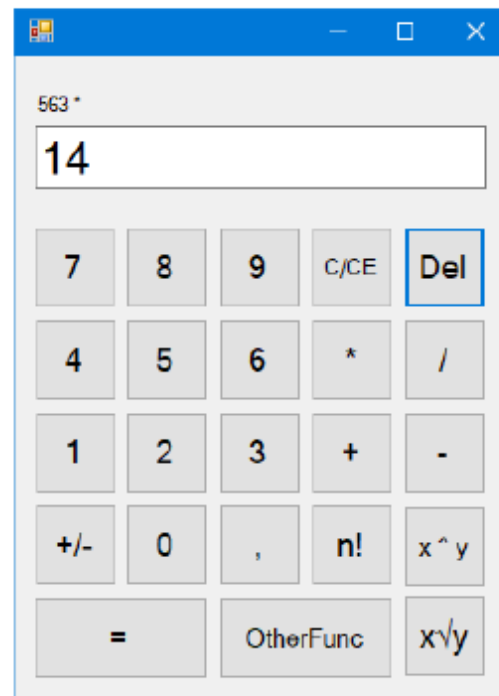
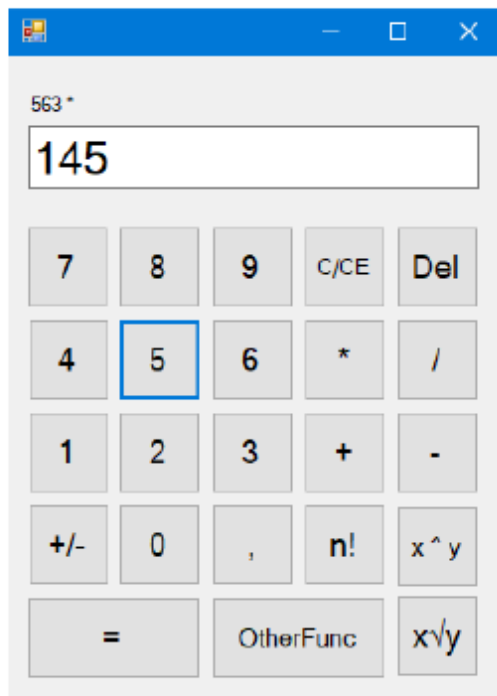
Pokud chcete smazat právě zadávaný výraz v zadávacím okně, klikněte 1x na tlačítko **C/CE**




Pokud chcete smazat úplně celý výraz, klikněte 2x na tlačítko **C/CE**



Tlačítko **Del** slouží k mazání právě zadávaného výrazu v zadávacím okně po jedné číslici zprava



- Microsoft VS Installer Projects
- Vytváří nové řešení (solution)
 - Není zahrnuto v odevzdávaném repozitáři



Microsoft Visual Studio Installer Projects

Microsoft | 📄 2,334,264 installs | ★★☆☆☆ (268) | Free

This official Microsoft extension provides support for Visual Studio Installer Projects in Visual Studio.

[Download](#)

- Větší dataset -> lepší výsledek
- 65 % času čtení a parsování vstupu
 - 5 % výpočet mocniny

zprava.md

4/25/2023

Profiling report

Profiling with dataset of 10 values

Profiling on small data set of just 10 numbers didn't show any significant bottleneck. Most of the time was spent in `Main` function, which is something we can very little about. Entire report from Visual Studio will be attached in folder *profiling*. Short summary of called functions is shown in the image below:

Function Name	Total CPU (unit, %)	Self CPU (unit, %)
Profiling.Program.Main(System.String[])	10 (17.21 %)	6 (10.34 %)
Profiling.StandardDeviation.SampleVariance(IEnumerable<double>)	2 (3.45 %)	1 (1.72 %)
Profiling.StandardDeviation.SumComponent(System.Collections.Generic.IEnumerable<double>)	1 (1.72 %)	1 (1.72 %)
External Call: System.Console.ReadLine()	1 (1.72 %)	1 (1.72 %)
External Call: System.Console.WriteLine()	1 (1.72 %)	1 (1.72 %)

Profiling with dataset of 10³ values

The situation is very similar to the previous one. The `Main` function is still the most time consuming one. Because there is no significant difference from the previous run, we will continue with profiling an even bigger data set.

Function Name	Total CPU (unit, %)	Self CPU (unit, %)
Profiling.Program.Main(System.String[])	9 (16.07 %)	5 (10.71 %)
Profiling.StandardDeviation.SampleVariance(IEnumerable<double>)	2 (3.47 %)	1 (1.79 %)
Profiling.StandardDeviation.SumComponent(System.Collections.Generic.IEnumerable<double>)	1 (1.79 %)	1 (1.79 %)
External Call: System.String.Split(IEnumerable<int>, System.StringSplitOptions)	1 (1.79 %)	1 (1.79 %)
Profiling.UtilsWrapper.ReadInput()	1 (1.79 %)	0 (0.00 %)

4/25/2023

significant impact on the
with section we can clearly

Function Name	Total CPU (unit, %)	Self CPU (unit, %)
Profiling.Program.Main(System.String[])	47 (46.18 %)	47 (46.18 %)
Profiling.StandardDeviation.SampleVariance(IEnumerable<double>)	14 (13.75 %)	14 (13.75 %)
Profiling.StandardDeviation.SumComponent(System.Collections.Generic.IEnumerable<double>)	14 (13.75 %)	14 (13.75 %)
External Call: System.String.Split(IEnumerable<int>, System.StringSplitOptions)	14 (13.75 %)	14 (13.75 %)
Profiling.UtilsWrapper.ReadInput()	14 (13.75 %)	14 (13.75 %)

in entire program were to
takes every number from
tely something worth

ed time. But this is not
Exponential().

Function Name	Total CPU (unit, %)	Self CPU (unit, %)
Profiling.Program.Main(System.String[])	10 (10.00 %)	10 (10.00 %)
Profiling.StandardDeviation.SampleVariance(IEnumerable<double>)	60 (60.00 %)	60 (60.00 %)
Profiling.StandardDeviation.SumComponent(System.Collections.Generic.IEnumerable<double>)	17 (17.00 %)	17 (17.00 %)
External Call: System.String.Split(IEnumerable<int>, System.StringSplitOptions)	1 (1.00 %)	1 (1.00 %)

ng reading and parsing of

not getting data from stdin,
(`ReadLine()`) function is also time

consuming, but if we want to check if the input is correct we have to use it. If we would 100% sure with input correctness, we could use just `Parse()`, but this will result in crash with incorrect input.

In calculation part, we could easily swap `MathLibrary.MathLib.Exponential(div, 2)` function for `MathLibrary.MathLib.Multiply(div, div)` which could improve performance

IVS Projekt - Kalkulačka

12/14

- Problém s komunikací
- Práce odvedena na poslední chvíli
- Nesoulad odvedených úkolů s plánem
- Výsledné chyby v důsledku integrace úkolů

