

Protokol k semestrální práci

Předmět: Úvod do programování a algoritmů

Zadání: S využitím Taylorova rozvoje vytvořte program pro výpočet přirozeného logaritmu.

Vypracoval: Jiří Ingr

Kruh: 159

Úvod

Zadáním úkolu je naprogramování Konzolové aplikace v programovacím jazyku C#, který vypočítá přibližnou hodnotu přirozeného logaritmu. Program umožňuje uživateli zadat jak hodnotu, ve které chce přirozený logaritmus vypočítat, tak počet iterací, neboli počet členů Taylorova rozvoje, od čehož se odvíjí výsledná přesnost výpočtu. Program byl napsán programem Microsoft Visual Studio 2019.

Řešení

Cílem bylo vytvořit přehledný program pro výpočet přibližné hodnoty logaritmu v zadaném bodě, který má ošetřené uživatelské vstupy a možnost opakování funkce programu po ukončení výpočtu.

Pro začátek jsem vypočítal několik členů Taylorova rozvoje přirozeného logaritmu a přišel na jednoduchost jeho výpočtu. Poté jsem začal převádět algoritmus do programovacího jazyka.

Nejprve jsem deklaroval vstupní proměnné a vytvořil možnost zadat uživateli proměnnou x (která určí bod, ve kterém se vypočítá hodnota $\ln x$) a počet členů (znamenantí pořadí nejvyššího vypočítaného členu rozvoje).

Ve výpočetní části jsem nejprve zajistil jednodušší a přesnější způsob výpočtu. V první kroku se omezí zadaná hodnota čísla tak, aby byla větší než 1 (pokud je číslo menší, převrátí se), což zapříčiní počítání pouze s kladnými čísly. Pro tuto část jsem vytvořil funkci *Prevraceni*, která zadané číslo převrátí nebo nechá v původním stavu. V dalším kroku jsem zadání podělil číslem e tolikrát, než byla jeho hodnota menší než \sqrt{e} , a v každém kroku přičetl jedničku do proměnné *pocetE*. V této části jsem použil for-cyklus.

Ze vzniklé hodnoty jsem již vypočítal Taylorův rozvoj pomocí funkce *Tayloruv_rozvoj*, v níž je implementována funkce *Tayloruv_clen*, které podle zadaných údajů vypočítají přibližnou hodnotu $\ln x$. K výsledku se nakonec přičte proměnná *pocetE*. Pokud bylo číslo převráceno, výsledku se změní znaménko. Následně se vypočítaná hodnota vypíše.

Nakonec jsem ošetřil uživatelské vstupy, jak ošetřením definičního oboru funkce, tak využitím podmínek *FormatException* (pro zadání „nezpracovatelné hodnoty“) a *Exception* (obecná podmínka) a vytvořil cyklus při dokončení výpočtu, který umožňuje vrátit se na začátek programu.

POZN.: Funkce *Tayloruv_clen*, *Tayloruv_rozvoj* a *Prevraceni* jsou součástí třídy *Funkce*.

Kód programu

Kód třídy program:

```
using System;

namespace Semestralni_konzolova_aplikace
{
    class Program
    {
```

```

static void Main(string[] args)
{
    // počáteční popis programu
    Console.ForegroundColor = ConsoleColor.DarkBlue;
    Console.WriteLine("Program na výpočet přibližné hodnoty funkce ln(x) v
daném bodě (pomocí taylorova rozvoje).\n");
    Console.ResetColor();

    do // cyklus pro opakování funkce programu, pokud uživatel zvolí možnost
dalšího výpočtu
    {
        // uživatelský vstup hodnoty x
        string vstupX = "";
        double x = 0;
        bool testVstupuX = true;
        while (testVstupuX)
        {
            // ošetření a přetypování vstupu hodnoty x
            try
            {
                Console.Write("Zadejte hodnotu x (x > 0): ");
                vstupX = Console.ReadLine();
                x = double.Parse(vstupX);
                // ošetření definičního oboru ln(x)
                if (x > 0)
                { testVstupuX = false; }
                else
                { // pokud je vstup mimo definiční obor ln(x) zobrazí se
chybová hláška
                    Console.ForegroundColor = ConsoleColor.Red;
                    Console.WriteLine("Pro tuto hodnotu není funkce ln(x)
definovaná. D(f) = (0; nekonečno)\nZadejte hodnotu z definičního oboru.");
                    Console.ResetColor();
                }
            }
            catch (FormatException) // ošetření formátu vstupu
            {
                Console.ForegroundColor = ConsoleColor.Red;
                Console.WriteLine("Hodna není ve správném formátu.\nZkuste to
znovu.");
                Console.ResetColor();
            }
            catch (Exception) // obecná podmínka
            {
                Console.ForegroundColor = ConsoleColor.Red;
                Console.WriteLine("Něco se nepovedlo.\nZkuste to znovu.");
                Console.ResetColor();
            }
        }
        Console.WriteLine(); // estetická mezera

        // uživatelský vstup počtu členů rozvoje
        string vstupPocetClenu = "";
        int pocetClenu = 0;
        bool testVstupuPocetClenu = true;
        while (testVstupuPocetClenu)
        {
            // ošetření a přetypování vstupu počtu členů
            try
            {
                Console.Write("Zadejte počet členů taylorova rozvoje: ");
                vstupPocetClenu = Console.ReadLine();

```

```

        pocetClenu = int.Parse(vstupPocetClenu);
        // ošetření hodnoty počtu členů rozvoje
        if (pocetClenu > 0)
            testVstupuPocetClenu = false;
        else
        { // chybová hláška se zobrazí, pokud uživatel zadá číslo <=
0
            Console.ForegroundColor = ConsoleColor.Red;
            Console.WriteLine("Neplatné číslo.\nZadejte celé číslo
větší než 0.");
            Console.ResetColor();
        }
    }
    catch (FormatException) // ošetření formátu vstupu
    {
        Console.ForegroundColor = ConsoleColor.Red;
        Console.WriteLine("Hodna není ve správném formátu.\nZadejte
celé číslo větší než 0.");
        Console.ResetColor();
    }
    catch (Exception) // obecná podmínka
    {
        Console.ForegroundColor = ConsoleColor.Red;
        Console.WriteLine("Něco se nepovedlo.\nZkuste to znovu.");
        Console.ResetColor();
    }
}

// potvrzení uživatelského vstupu
Console.ForegroundColor = ConsoleColor.Yellow;
Console.WriteLine("\nVypočítám pomocí {0} členů taylorova rozvoje
hodnotu ln({1}).", pocetClenu, x);
Console.ResetColor();

// pokračování k výpočtu (aby si uživatel mohl zkontrolovat své
zadání)
Console.WriteLine("Pro pokračování stikněte Enter...\n");
while (Console.ReadKey().Key != ConsoleKey.Enter) { }

// VÝPOČET
// pokud x náleží (0; 1) převrátí se jeho hodnota
double mezivypocet1 = Funkce.Prevraceni(x);
bool prevraceno = true;
if (mezivypocet1 == x)
    prevraceno = false;

// proměnná x se dělí eulerovým číslem, až než je menší než jeho
hodnota
int pocetE = 0;
double e = Math.Exp(1);
double odmocninaE = 1.6488; // odmocnina z e
while (mezivypocet1 > odmocninaE)
{
    mezivypocet1 /= e;
    pocetE++;
}

// výpočet přibližné hodnoty pomocí taylorova rozvoje
double mezivypocet2 = Funkce.Tayloruv_rozvoj(mezivypocet1,
pocetClenu);

// k mezivýsledku se přičte proměnná pocetE

```

```

        double mezivypocet3 = mezivypocet2 + pocetE;

        // výsledek se převrátí zpět v případě, že se výše jeho hodnota
převrátila
        double vysledek;
        if (prevraceno)
            vysledek = -mezivypocet3;
        else
            vysledek = mezivypocet3;

        // vypsání hodnoty uživateli
        Console.WriteLine("Hodnota ln({0}) s danou přesností je přibližně: ",
vstupX);

        Console.ForegroundColor = ConsoleColor.Green;
        Console.WriteLine(vysledek);
        Console.ResetColor();

        // příkaz pro ukončení programu klávesou Esc
        Console.WriteLine("\nPro ukončení programu stiskni klávesu Esc, Nový
výpočet spustíte stisknutím libovolné jiné klávesy...\n");
    } while (Console.ReadKey().Key != ConsoleKey.Escape);
}
}
}

```

Kód třídy funkce:

```

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

namespace Semestralni_konzolova_aplikace
{
    class Funkce
    {
        // vypocet taylorova clenou
        private static double Tayloruv_clen (double promenna, int poradiCleny) //
vstupem je vstupní hodnota x a pořadí členu rozvoje
        { // výpočet čitatele členu taylorova rozvoje
            double citatel = 1;
            for (int i = 0; i < poradiCleny; i++)
            {
                citatel = citatel * (promenna - 1);
            }
            // výpočet jmenovatele členu taylorova rozvoje
            int jmenovatel = poradiCleny;
            double clen = citatel / jmenovatel;
            // znaménko členu (ale se každý lichý člen přičetl a každý sudý odečetl)
            if (poradiCleny % 2 == 0)
                clen = -clen;
            return clen; // vrací hodnotu jednoho členu taylorova rozvoje
        }

        // vypocet taylorova rozvoje
        public static double Tayloruv_rozvoj(double promenna, int pocetCleny)
        {
            double rozvoj = 0;
            for (int j = 0; j < pocetCleny; j++) // cyklus pro přičítání jednotlivých
členu taylorova rozvoje
            {
                rozvoj = rozvoj + Tayloruv_clen(promenna, j + 1); // funkce
Tayloruv_clen výše
            }
        }
    }
}

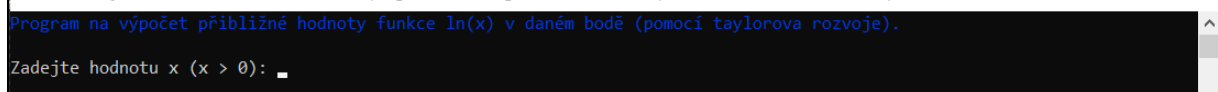
```

```
    }  
    return rozvoj; // vrátí výslednou hodnotu taylorova rozvoje vypočítaného  
na zadaný počet iterací  
}  
  
// prevracení hodnoty pokud je hodnota menší než 1  
public static double Prevraceni(double promenna)  
{  
    if (promenna < 1) // pokud je hodnota menší než nula, převrátí se  
    {  
        promenna = 1 / promenna;  
    }  
    else { } // pokud je větší než 0, zůstane stejná  
    return promenna; // metoda vrací stejnou vstupní hodnotu, nebo převrácenou  
vstupní hodnotu  
}  
}  
}
```

Ukázka programu v Console Application

Ukázka základních funkcí programu:

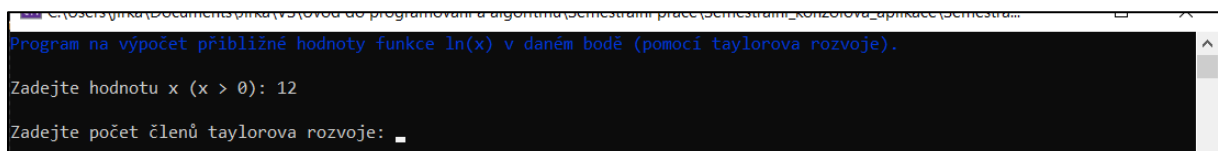
Program nejprve vypíše text s informací o svojí funkci a vybídne uživatele k zadání prvního vstupu, kterým je hodnota x pro výpočet $\ln x$ (viz obr. 1):



Program na výpočet přibližné hodnoty funkce $\ln(x)$ v daném bodě (pomocí taylorova rozvoje).
Zadejte hodnotu x ($x > 0$):

obr. 1 - Vstup 1

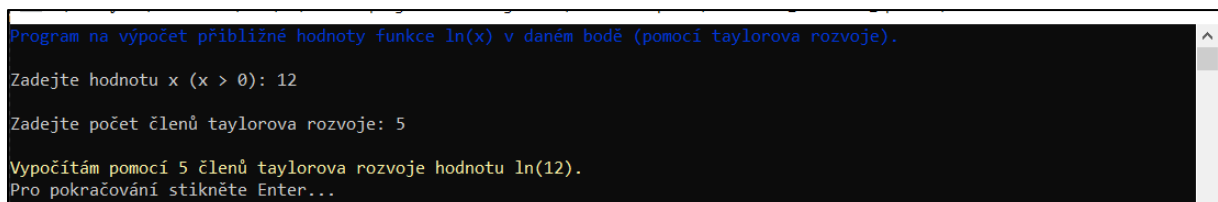
Dále program vyžaduje zadání druhého uživatelského vstupu s přesností výpočtu (viz obr. 2):



Program na výpočet přibližné hodnoty funkce $\ln(x)$ v daném bodě (pomocí taylorova rozvoje).
Zadejte hodnotu x ($x > 0$): 12
Zadejte počet členů taylorova rozvoje:

obr. 2 - Vstup 2

Po zadání vstupů dojde k vypsání zadaných hodnot. Po překontrolování uživatel pokračuje stisknutím klávesy Enter (viz obr. 3):



Program na výpočet přibližné hodnoty funkce $\ln(x)$ v daném bodě (pomocí taylorova rozvoje).
Zadejte hodnotu x ($x > 0$): 12
Zadejte počet členů taylorova rozvoje: 5
Vypočítám pomocí 5 členů taylorova rozvoje hodnotu $\ln(12)$.
Pro pokračování stikněte Enter...

obr. 3 - Kontrola vstupu

Nakonec se vypíše výsledek výpočtu zvýrazněný zelenou barvou. Uživatel má možnost program ukončit klávesou Esc nebo pokračovat dalším výpočtem (viz obr. 4):

```
Program na výpočet přibližné hodnoty funkce ln(x) v daném bodě (pomocí taylorova rozvoje).  
Zadejte hodnotu x (x > 0): 12  
Zadejte počet členů taylorova rozvoje: 5  
Vypočítám pomocí 5 členů taylorova rozvoje hodnotu ln(12).  
Pro pokračování stikněte Enter...  
Hodnota ln(12) s danou přesností je přibližně: 2,4913359478716623  
Pro ukončení programu stiskni klávesu Esc, Nový výpočet spustíte stiknutím libovolné jiné klávesy...
```

obr. 4 - Výsledek výpočtu

Ukázka funkce programu při zadání velmi vysoké a velmi nízké hodnoty x:

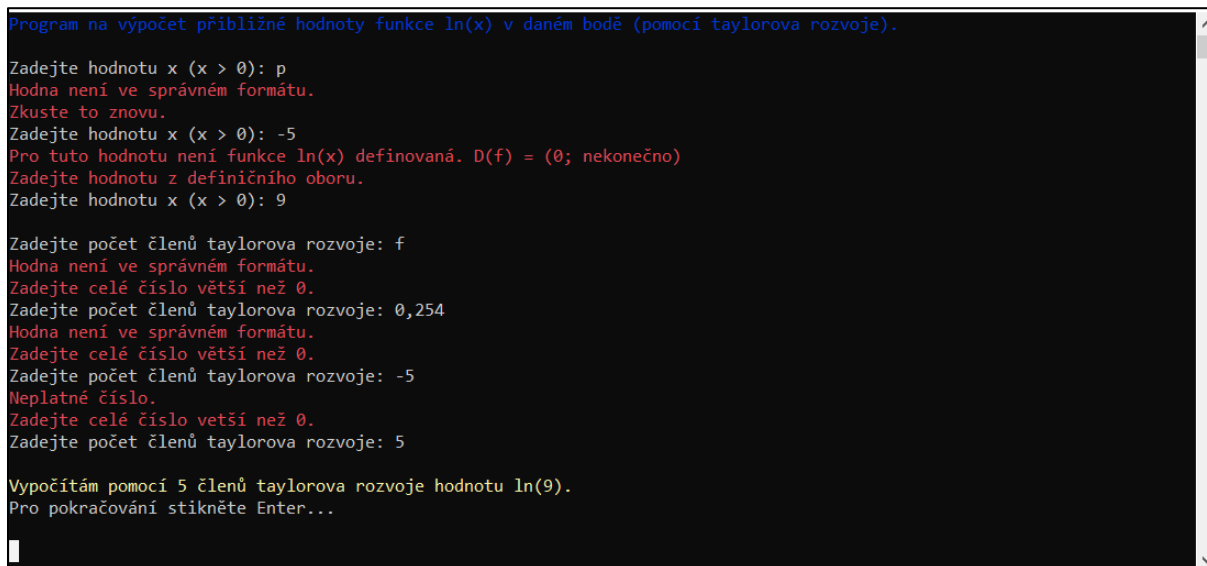
Program funguje při zadání libovolně vysoké či nízké hodnoty v definičním obor přirozeného logaritmu (viz obr. 5):

```
Program na výpočet přibližné hodnoty funkce ln(x) v daném bodě (pomocí taylorova rozvoje).  
Zadejte hodnotu x (x > 0): 9876  
Zadejte počet členů taylorova rozvoje: 70  
Vypočítám pomocí 70 členů taylorova rozvoje hodnotu ln(9876).  
Pro pokračování stikněte Enter...  
Hodnota ln(9876) s danou přesností je přibližně: 9,19786285046507  
Pro ukončení programu stiskni klávesu Esc, Nový výpočet spustíte stiknutím libovolné jiné klávesy...  
Zadejte hodnotu x (x > 0): 0,00056  
Zadejte počet členů taylorova rozvoje: 70  
Vypočítám pomocí 70 členů taylorova rozvoje hodnotu ln(0,00056).  
Pro pokračování stikněte Enter...  
Hodnota ln(0,00056) s danou přesností je přibližně: -7,487573774235079  
Pro ukončení programu stiskni klávesu Esc, Nový výpočet spustíte stiknutím libovolné jiné klávesy...
```

obr. 5 - Extrémní hodnoty

Ukázka chybových hlášek při zadání nevhodných vstupů:

Program má ošetřeny také chybné vstupy, po jejichž zadání se zobrazují chybové hlášky (viz obr. 6):



```
Program na výpočet přibližné hodnoty funkce ln(x) v daném bodě (pomocí taylorova rozvoje).  
Zadejte hodnotu x (x > 0): p  
Hodna není ve správném formátu.  
Zkuste to znovu.  
Zadejte hodnotu x (x > 0): -5  
Pro tuto hodnotu není funkce ln(x) definovaná. D(f) = (0; nekonečno)  
Zadejte hodnotu z definičního oboru.  
Zadejte hodnotu x (x > 0): 9  
  
Zadejte počet členů taylorova rozvoje: f  
Hodna není ve správném formátu.  
Zadejte celé číslo větší než 0.  
Zadejte počet členů taylorova rozvoje: 0,254  
Hodna není ve správném formátu.  
Zadejte celé číslo větší než 0.  
Zadejte počet členů taylorova rozvoje: -5  
Neplatné číslo.  
Zadejte celé číslo větší než 0.  
Zadejte počet členů taylorova rozvoje: 5  
  
Vypočítám pomocí 5 členů taylorova rozvoje hodnotu ln(9).  
Pro pokračování stikněte Enter...
```

obr. 6 - Chybové hlášky

Závěr

Úkol pro mě byl zajímavou ukázkou toho, jak počítače vyhodnocují hodnoty složitějších funkcí, a následným zjištěním, že se nejedná o zdaleka tak složitý algoritmus, jaký jsem předpokládal. Po vypočítání několika členů Taylorova rozvoje logaritmu jsem odvodil jeho jednoduchost a naprogramování už šlo snadno. Nejvíce mě bavilo postupné zjednodušování funkce algoritmu i když místy jsem narážel zejména na syntaktické problémy. Všechny problémy jsem ale nakonec zdárně vyřešil.