

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace \_10\_Matematik\_Metotlari

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

//Sabit değişken(field) tanımlamak için aşağıdaki şekilde kullanım geçerlidir. herhangi bir tipte sabit değişken tanımlanabilir.

//const <tip adi> <değişken adı> = <degeri>

const double kendiPi = 3.14;

//Error 1 A const field requires a value to be provided => tanımlandığı an değişkene değeri atanmak zorundadır.

//const double kendiPi2;

private void Form1\_Load(object sender, EventArgs e)

{

//Sabit tanımlanmış bir öğe değiştirilemez(set).

//kendiPi = 3.15;

}

private void btnMutlakDeger\_Click(object sender, EventArgs e)

{

int sayi = int.Parse(txtSayi1.Text);

//if (sayi < 0)

// sayi \*= -1;

//Aşağıdaki kullanımda sayi havada kalır. Yani herhangi birşeye atanmaz ise sayinin mutlak değerini almış olmam...

//Math.Abs(sayi);

sayi = Math.Abs(sayi);

txtSayi2.Text = sayi.ToString();

}

private void btnPi\_Click(object sender, EventArgs e)

{

txtSayi1.Text = Math.PI.ToString();

}

private void btnBuyukSayi\_Click(object sender, EventArgs e)

{

int sayi1 = int.Parse(txtSayi1.Text);

int sayi2 = int.Parse(txtSayi2.Text);

int buyukSAyi = Math.Max(sayi1, sayi2);

this.Text = String.Format("{0} sayısı daha büyüktür", buyukSAyi);

}

private void btnKucukSayi\_Click(object sender, EventArgs e)

{

int sayi1 = int.Parse(txtSayi1.Text);

int sayi2 = int.Parse(txtSayi2.Text);

int kucukSAyi = Math.Min(sayi1, sayi2);

this.Text = String.Format("{0} sayısı daha küçüktür", kucukSAyi);

}

private void btnAsagiYuvarla\_Click(object sender, EventArgs e)

{

double ondalikSayi = double.Parse(txtSayi1.Text);

double sonuc = Math.Floor(ondalikSayi);

this.Text = String.Format("{0} sayısının aşağı yuvarlanmış hali: {1}", ondalikSayi, sonuc);

}

private void btnYukariYuvarla\_Click(object sender, EventArgs e)

{

double ondalikSAyi = double.Parse(txtSayi1.Text);

double sonuc = Math.Ceiling(ondalikSAyi);

this.Text = String.Format("{0} sayısının yukarı yuvarlanmış hali: {1}", ondalikSAyi, sonuc);

}

private void btnYuvarla\_Click(object sender, EventArgs e)

{

double ondalikSayi = double.Parse(txtSayi1.Text);

double sonuc = Math.Round(ondalikSayi, MidpointRounding.ToEven);

txtSayi2.Text = sonuc.ToString();

//AwayFromZero : Eğer sayının ondalıklı kısmı 5 ve 5ten büyükse bir üst sayıya yuvarlar. Aksi durumda ondalıklı kısmı atıp o şekilde sayıyı göndürür.

//ToEven : Eğer sayının ondalıklı kısmı 5 şeklinde bitiyorsa, sayı her zaman çifte tamamlanır.

}

private void btnKareKok\_Click(object sender, EventArgs e)

{

double sayi = double.Parse(txtSayi1.Text);

double karekok = Math.Sqrt(sayi);

txtSayi2.Text = karekok.ToString();

}

private void btnUsAlma\_Click(object sender, EventArgs e)

{

double sayi = double.Parse(txtSayi1.Text);

double kati = double.Parse(txtSayi2.Text);

double usAlinmisHali = Math.Pow(sayi,kati);

this.Text = usAlinmisHali.ToString();

}

private void btnE\_Click(object sender, EventArgs e)

{

txtSayi1.Text = Math.E.ToString();

}

}

}