

 public partial class \_2\_while : Form

 {

 public \_2\_while()

 {

 InitializeComponent();

 }

 private void \_2\_while\_Load(object sender, EventArgs e)

 {

 /\*

 while döngüsü:

 Döngüler arasındaki en basit yapıya sahip olandır. Sadece tek bir bölümden oluşur, oda kontrol edilecek bir koşuldur. Yani içerisinde geriye bool (true/false) döndüren bir ifade ister. True döndüğü sürece çalışır.

 \*/

 int sayac = 0;

 while (sayac < 10)

 {

 //MessageBox.Show("Hello World " + sayac);

 sayac++;

 }

 }

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

 {

 listBox1.Items.Clear();

 int sayac = 1;

 while (sayac <= 100)

 {

 listBox1.Items.Add(sayac);

 sayac++;

 }

 }

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

 {

 listBox1.Items.Clear();

 int sayac = 100;

 while (sayac >= 1)

 {

 listBox1.Items.Add(sayac);

 sayac--;

 }

 }

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

 {

 listBox1.Items.Clear();

 int sayac = 0;

 while (sayac <= 100)

 {

 if (sayac % 2 == 0)

 {

 listBox1.Items.Add(sayac);

 }

 sayac++;

 }

 }

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

 {

 int sayi = 0;

 while (sayi < 101)

 {

 sayi += 5;

 if (sayi == 15)

 continue;

 listBox1.Items.Add(sayi);

 }

 }

 private void btnBirYuz33denSonrasiYok\_Click(object sender, EventArgs e)

 {

 listBox1.Items.Clear();

 int sayac = 0;

 while (sayac < 100)

 {

 listBox1.Items.Add(sayac);

 if (sayac == 33)

 break;

 sayac++;

 }

 }

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

 {

 listBox1.Items.Clear();

 Random rnd = new Random();

 int i = 0;

 while (i < 10)

 {

 int rastgele = rnd.Next(25);

 if (listBox1.Items.IndexOf(rastgele) == -1)

 {

 listBox1.Items.Add(rastgele);

 }

 else

 {

 i--;

 }

 i++;

 }

 listBox1.Sorted = true;

 }

 private void btnRastgeleSayilar2\_Click(object sender, EventArgs e)

 {

 listBox1.Items.Clear();

 Random rnd = new Random();

 while (listBox1.Items.Count < 10)

 {

 int rastgele = rnd.Next(25);

 if (listBox1.Items.IndexOf(rastgele) == -1)

 listBox1.Items.Add(rastgele);

 listBox1.Sorted = true;



public partial class \_3\_do\_while : Form

 {

 public \_3\_do\_while()

 {

 InitializeComponent();

 }

 private void \_3\_do\_while\_Load(object sender, EventArgs e)

 {

 /\*

 do while döngüsü: tıpkı while döngüsü gibi kendisine verilen parametre true döndüğü sürece çalışan döngü çeşitidir. while döngüsünden tek farkı, en az bir kere döngünün çalışıyor olması.

 \*/

 }

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

 {

 int sayi = 10;

 do

 {

 listBox1.Items.Add(sayi);

 sayi++;

 }

 while (sayi < 15);

 }

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

 {

 timer1.Enabled = true;

 }

 private void timer1\_Tick(object sender, EventArgs e)

 {

 progressBar1.Value++;

 if (progressBar1.Value == 18)

 {

 timer1.Enabled = false;

 DialogResult sonuc;

 do

 {

 sonuc = MessageBox.Show("Windows işletim sistemi mcs.dll'i bulamadı. Bu yüzden bilgisayarınız kapanacaktır.", "Hata", MessageBoxButtons.AbortRetryIgnore, MessageBoxIcon.Error);

 }

 while (sonuc == System.Windows.Forms.DialogResult.Retry);

 }

 }



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 \_9\_Donguler

{

 public partial class \_4\_foreach : Form

 {

 public \_4\_foreach()

 {

 InitializeComponent();

 }

 private void \_4\_foreach\_Load(object sender, EventArgs e)

 {

 /\*

 foreach döngüsü:

 Kendisine verilen bir koleksiyondaki (dizi) tüm elemanları tek tek elde etmek amacıyla kullanılan bir döngü çeşitidir. foreach döngüsünün çalışma prensibi gereği, döngü sırasında, geriye doğru gidilmez. Yani foreach koleksiyonu elemanları baştan başlayarak ve sürekli ileriye doğru tek tek elde etmek için kullanılır ve bu şekilde bir çalışma yapısı vardır. Yine çalışma mantığı gereği tek tek elde edilen elemanlar değiştirilemez, yeni bir değer atanamaz. Sadece okunabilir (read-only)

 FORO => Forward Only - Read Only

 foreach 3 kısımdan oluşur:

 1- Elemanları tek tek elde edilecek olan koleksiyon/dizi

 2- Bu koleksiyondan elde edilecek elemanların hangi tipte yakalanacağı

 3- Belirtilen tipte yakalanan elemanların hangi isim ile ulaşılacağı

 \*/

 string[] sepet = { "iPhone 4S", "Samsun Galaxy S2", "Nokia 3310" };

 foreach (string urun in sepet)

 {

 MessageBox.Show(urun);

 }

 }

 string[] sinif = new string[] { "Kaan", "Zeynep", "Celil", "Arda", "Oğuzhan", "Ahmet", "Mustafa", "Hakan", "Alper", "Taha" };

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

 {

 listBox1.Items.Clear();

 for (int i = 0; i < sinif.Length; i++)

 {

 listBox1.Items.Add(sinif[i]);

 }

 }

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

 {

 listBox1.Items.Clear();

 foreach (string ogrenci in sinif)

 {

 listBox1.Items.Add(ogrenci);

 }

 }

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

 {

 foreach (Control kontrol in this.Controls)

 {

 kontrol.BackColor = Color.Black;

 }

 }

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

 {

 foreach (Control kontrol in this.Controls)

 {

 if (kontrol is Button)

 {

 kontrol.BackColor = Color.Blue;

 }

 }

 }

 }

}