****

**T. C.**

**MANİSA CELAL BAYAR ÜNİVERSİTESİ**

**HASAN FERDİ TURGUTLU TEKNOLOJİ FAKÜLTESİ**

**YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ**

**Yazılım Yapımı Vize – Final Ödev Raporu**

**Mert Oğuz Dalak – 182803037**

**Dr. Öğr. Üyesi Emin BORANDAĞ**

**MANİSA**

**2020**

Çözüm Mantığı (İşlem Kısmı – Vize)

Kodumuz textbox üzerinden erişebildiği sayıları, araştırmalarım sonucu bu kod için uygun olan genetik algoritma kullanarak işlemden geçirir. Buradaki amaç hedef sayıyı veya ona en yakın sayıyı bulana kadar kodlarımızı işlemden geçirmektir.

Kaynak Kodlar (İşlem Kısmı – Vize)

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace yazilim\_yapimi\_islem\_kismi

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

int toplam\_puan = 0;

int puan = 10;

int gecicipuan;

Random rnd = new Random();

public static double secondsTaken = 0;

public static int bestSoFar = 0;

private void Hesaplama()

{

int[] teksayi = new int[6];

teksayi[0] = Convert.ToInt32(txt\_sayi1.Text);

teksayi[1] = Convert.ToInt32(txt\_sayi2.Text);

teksayi[2] = Convert.ToInt32(txt\_sayi3.Text);

teksayi[3] = Convert.ToInt32(txt\_sayi4.Text);

teksayi[4] = Convert.ToInt32(txt\_sayi5.Text);

teksayi[5] = Convert.ToInt32(txt\_sayi6.Text);

int hedefsayi = Convert.ToInt32(txt\_hedef.Text);

DateTime startTime = DateTime.Now;

while (true)

{

secondsTaken = (DateTime.Now - startTime).TotalSeconds;

int num = rnd.Next(6);

int temp = teksayi[num];

teksayi[num] = teksayi[0];

teksayi[0] = temp;

int thisTotal = teksayi[0];

string solution = teksayi[0].ToString();

int numbersToUse = rnd.Next(1, 7);

for (int i = 1; i < numbersToUse; i++)

{

int operation = rnd.Next(4);

if (operation == 0)

{

thisTotal += teksayi[i];

solution += " + " + teksayi[i].ToString();

}

if (operation == 1)

{

thisTotal -= teksayi[i];

solution += " - " + teksayi[i].ToString();

}

if (operation == 2)

{

thisTotal \*= teksayi[i];

solution += " x " + teksayi[i].ToString();

}

if (operation == 3)

{

if (thisTotal % teksayi[i] != 0) continue;

thisTotal /= teksayi[i];

solution += " / " + teksayi[i].ToString();

}

}

solution += " = " + thisTotal.ToString();

if (secondsTaken >= 10)

{

if (Math.Abs(hedefsayi - bestSoFar) < 10)

{

gecicipuan = puan - Math.Abs(hedefsayi - bestSoFar);

txt\_kpuan.Text = Convert.ToString(gecicipuan);

toplam\_puan += gecicipuan;

txt\_toplampuan.Text = Convert.ToString(toplam\_puan);

break;

}

else

{

MessageBox.Show("Çözüm Bulunamadı");

break;

}

}

if (Math.Abs(hedefsayi - thisTotal) < Math.Abs(hedefsayi - bestSoFar))

{

bestSoFar = thisTotal;

txt\_islem.Text = solution;

}

if (thisTotal == hedefsayi)

{

txt\_islem.Text = solution;

txt\_kpuan.Text = Convert.ToString(puan);

toplam\_puan += puan;

txt\_puan.Text = Convert.ToString(toplam\_puan);

break;

}

}

txt\_islemsuresi.Text = Convert.ToString(secondsTaken);

}

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

{

}

private void TextEnable(bool a)

{

txt\_sayi6.Enabled = a;

txt\_sayi1.Enabled = a;

txt\_sayi2.Enabled = a;

txt\_sayi3.Enabled = a;

txt\_sayi4.Enabled = a;

txt\_sayi5.Enabled = a;

txt\_hedef.Enabled = a;

if (a == true)

{

txt\_sayi6.Clear();

txt\_sayi1.Clear();

txt\_sayi2.Clear();

txt\_sayi3.Clear();

txt\_sayi4.Clear();

txt\_sayi5.Clear();

txt\_hedef.Clear();

}

}

private void rdo\_random\_CheckedChanged(object sender, EventArgs e)

{

if (rdo\_random.Checked == true)

{

TextEnable(false);

int[] randomsayiuret = new int[6];

for (int i = 1; i < 6; i++)

{

randomsayiuret[i] = rnd.Next(1, 10);

}

txt\_sayi1.Text = randomsayiuret[1].ToString();

txt\_sayi2.Text = randomsayiuret[2].ToString();

txt\_sayi3.Text = randomsayiuret[3].ToString();

txt\_sayi4.Text = randomsayiuret[4].ToString();

txt\_sayi5.Text = randomsayiuret[5].ToString();

int randomciftsayiuret = rnd.Next(10, 99);

randomciftsayiuret = randomciftsayiuret - randomciftsayiuret % 10;

txt\_sayi6.Text = randomciftsayiuret.ToString();

int randomhedef = rnd.Next(100, 999);

txt\_hedef.Text = randomhedef.ToString();

}

txt\_islem.Text = " ";

}

private void rdo\_kendingir\_CheckedChanged(object sender, EventArgs e)

{

if (rdo\_kendingir.Checked == true)

{

TextEnable(true);

}

txt\_islem.Text = " ";

}

private void btn\_hesapla\_Click(object sender, EventArgs e)

{

if (txt\_hedef.Text == "")

{

MessageBox.Show("Lütfen Sayıları Boş Bırakmayınız!");

}

else

{

Hesaplama();

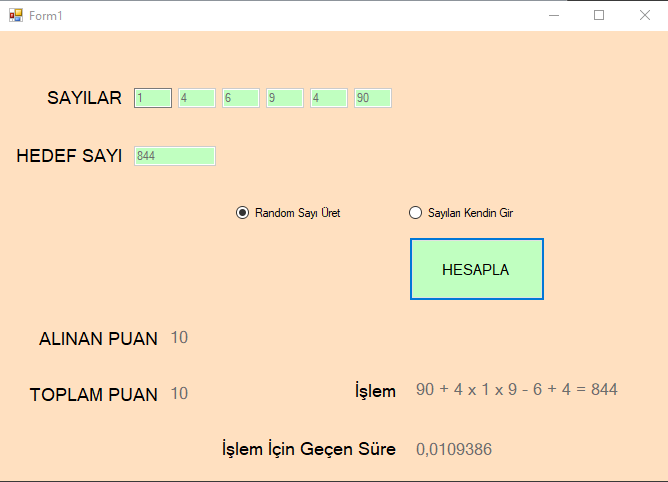
}

}

}

}

Çalışır Ekran Görüntüsü (İşlem Kısmı – Vize)



Çözüm Mantığı (Kelime Kısmı – Final)

1-Textbox’tan eriştiğimiz harfleri bir kelime dizisinin içinde tutuyoruz

2-Sql’den kelimeleri çekerken önce en uzun kelimeden en kısa kelimeye doğru sıralatıyoruz

2-Sql’den çektiğimiz kelimeleri harflerine ayırarak char dizisi haline getirip kelime dizimizin içindeki harflerle uyumluluğunu kontrol ettiriyoruz

3-Eğer çekilen kelimenin harfleri kelime dizisindeki elemanlarla eşleşiyorsa ya da 1 fazlasıysa ya da kelime dizisindeki elemanların sayısından az ise o kelimeyi bir değişkene aktarıp ekrana yazdırıyoruz.

4- 2. Maddede bahsedilen sıralama işlemi sayesinde ilk bulduğu kelimeden sonra kod kelime çekme işlemini durdurup en uzun kelimeyi en kısa sürede yazdırabilmiş oluyor.

Kaynak Kodlar (Kelime Kısmı – Final)

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace KelimeBul

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

SqlConnection n = new SqlConnection("Server=.\\SQLEXPRESS;Database=BirKelime;trusted\_connection=true;");

DataSet daset = new DataSet();

public static double secondsTaken = 0;

string kelimelistesi;

int a;

int b;

private void rdo\_rastgele\_CheckedChanged(object sender, EventArgs e)

{

Random rastgele = new Random();

Random rastgele2 = new Random();

string[] uret = new string[9];

string sesliharfler = "aeıioöuü";

string sessizharfler = "bcdfgğhjklmnprsştvyz";

for (int i = 0; i < 8; i++)

{

uret[i] = "";

}

for (int x = 0; x < 4; x++)

{

uret[x] = sesliharfler[rastgele.Next(sesliharfler.Length)].ToString();

}

for (int j = 4; j < 8; j++)

{

uret[j] = sessizharfler[rastgele2.Next(sessizharfler.Length)].ToString();

}

txt\_bir.Text = uret[0];

txt\_iki.Text = uret[1];

txt\_uc.Text = uret[2];

txt\_dort.Text = uret[3];

txt\_bes.Text = uret[4];

txt\_alti.Text = uret[5];

txt\_yedi.Text = uret[6];

txt\_sekiz.Text = uret[7];

txt\_joker.Text = "A-Z";

txt\_bulunankelime.Text = " ";

}

private void rdo\_kendingir\_CheckedChanged(object sender, EventArgs e)

{

txt\_bir.Text = "";

txt\_iki.Text = "";

txt\_uc.Text = "";

txt\_dort.Text = "";

txt\_bes.Text = "";

txt\_alti.Text = "";

txt\_yedi.Text = "";

txt\_sekiz.Text = "";

txt\_joker.Text = "A-Z";

txt\_bulunankelime.Text = " ";

}

private void btn\_hesapla\_Click(object sender, EventArgs e)

{

KelimeBul();

}

private void KelimeBul()

{

DateTime startTime = DateTime.Now;

while (true)

{

secondsTaken = (DateTime.Now - startTime).TotalSeconds;

SqlCommand comm = new SqlCommand("SELECT F1 FROM Sayfa1$ ORDER BY LEN(F1) DESC", n);

n.Open();

SqlDataReader dr = comm.ExecuteReader();

string[] ayrilmisharf = new string[9];

while (dr.Read())

{

ayrilmisharf[0] = txt\_bir.Text;

ayrilmisharf[1] = txt\_iki.Text;

ayrilmisharf[2] = txt\_uc.Text;

ayrilmisharf[3] = txt\_dort.Text;

ayrilmisharf[4] = txt\_bes.Text;

ayrilmisharf[5] = txt\_alti.Text;

ayrilmisharf[6] = txt\_yedi.Text;

ayrilmisharf[7] = txt\_sekiz.Text;

a = 0;

kelimelistesi = Convert.ToString(dr["F1"]);

char[] harflereayir = kelimelistesi.ToCharArray();

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

{

if (Array.IndexOf(ayrilmisharf, harflereayir[i].ToString()) == -1)

{

b = i;

a++;

}

else

{

for (int y = 0; y < 8; y++)

{

if (ayrilmisharf[y] == harflereayir[i].ToString())

{

ayrilmisharf[y] = " ";

break;

}

}

}

}

if (a == 0)

{

ayrilmisharf[8] = harflereayir[b].ToString();

txt\_bulunankelime.Text = kelimelistesi;

puanhesapla();

break;

}

else if (a == 1)

{

ayrilmisharf[8] = harflereayir[b].ToString();

txt\_joker.Text = ayrilmisharf[8];

txt\_bulunankelime.Text = kelimelistesi;

puanhesapla();

break;

}

}

dr.Close();

n.Close();

secondsTaken = (DateTime.Now - startTime).TotalSeconds;

break;

}

txt\_sure.Text = secondsTaken.ToString();

}

int kelimepuan = 0;

int toppuan = 0;

private void puanhesapla()

{

for (int i = 9; i >= 0; i--)

{

if (txt\_bulunankelime.Text.Length == i)

{

if (i == 9)

{

kelimepuan = 15;

}

else if (i == 8)

{

kelimepuan = 11;

}

else if (i == 7)

{

kelimepuan = 9;

}

else if (i == 6)

{

kelimepuan = 7;

}

else if (i == 5)

{

kelimepuan = 5;

}

else if (i == 4)

{

kelimepuan = 4;

}

else if (i == 3)

{

kelimepuan = 3;

}

txt\_kelimepuan.Text = kelimepuan.ToString();

toppuan += kelimepuan;

txt\_toplampuan.Text = toppuan.ToString();

}

}

}

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

{

}

private void txt\_toplampuan\_TextChanged(object sender, EventArgs e)

{

}

}

}

Çalışır Ekran Görüntüsü (Kelime Kısmı – Final)

