****

**T. C.**

**MANİSA CELAL BAYAR ÜNİVERSİTESİ**

**HASAN FERDİ TURGUTLU TEKNOLOJİ FAKÜLTESİ**

**YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ**

**YAZILIM YAPIMI**

**BİR KELİME BİR İŞLEM PROJESİ**

**Ömer Furkan BİLGE 182803052**

**Samed KAZANIR 182803053**

**Dersin Sorumlu Öğretim Üyesi**

**Dr.Öğr.Üyesi Emin BORANDAĞ**

**MANİSA**

**2020**

**1) BİR İŞLEM-VİZE**

**Çözüm Mantığı**

Kodun temel çalışma prensibi, kullanıcıdan manuel alınarak veya random olarak üretilen sayıların, genetik algoritma kullanılarak bir dizi işlemden geçirilip mümkünse hedef sayı değerini, değilse en yakın sayı değerini çözüm olarak kullanıcıya sunar. Genetik algoritma mantığını kısaca özetlemek gerekirse; elimizdeki sayıları önce dizide karıştırıp, sonra aynı şekilde karışık olarak istenen işlemleri yaparak, kendini sürekli güncelleyip doğru sonuca ulaşıncaya kadar bu işlemi devam ettirir. Biz de projemizde bu algoritma mantığını kullanmayı uygun gördük.

**Kaynak Kod**

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 Yazılım\_Yapmı\_Ödev\_1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

int c;

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(mtxt\_tek1.Text);

teksayi[1] = Convert.ToInt32(mtxt\_tek2.Text);

teksayi[2] = Convert.ToInt32(mtxt\_tek3.Text);

teksayi[3] = Convert.ToInt32(mtxt\_tek4.Text);

teksayi[4] = Convert.ToInt32(mtxt\_tek5.Text);

teksayi[5] = Convert.ToInt32(mtxt\_cift.Text);

int hedefsayi = Convert.ToInt32(mtxt\_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\_cozum.Text = solution;

}

if (thisTotal == hedefsayi)

{

txt\_cozum.Text = solution;

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

toplam\_puan += puan;

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

break;

}

}

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

}

private int Kontrol()

{

c = 0;

if (mtxt\_tek1.Text.Length < 1 || Convert.ToInt32(mtxt\_tek1.Text) < 1)

{

MessageBox.Show("1.Sayıyı Hatalı Girdiniz");

c++;

}

if (mtxt\_tek2.Text.Length < 1 || Convert.ToInt32(mtxt\_tek2.Text) < 1)

{

MessageBox.Show("2.Sayıyı Hatalı Girdiniz");

c++;

}

if (mtxt\_tek3.Text.Length < 1 || Convert.ToInt32(mtxt\_tek3.Text) < 1)

{

MessageBox.Show("3.Sayıyı Hatalı Girdiniz");

c++;

}

if (mtxt\_tek4.Text.Length < 1 || Convert.ToInt32(mtxt\_tek4.Text) < 1)

{

MessageBox.Show("4.Sayıyı Hatalı Girdiniz");

c++;

}

if (mtxt\_tek5.Text.Length < 1 || Convert.ToInt32(mtxt\_tek5.Text) < 1)

{

MessageBox.Show("5.Sayıyı Hatalı Girdiniz");

c++;

}

if (mtxt\_cift.Text.Length != 2 || Convert.ToInt32(mtxt\_cift.Text) < 10)

{

MessageBox.Show("6.Sayıyı Hatalı Girdiniz");

c++;

}

else if (Convert.ToInt32(mtxt\_cift.Text) % 10 != 0)

{

MessageBox.Show("6.Sayı 10 veya 10`un Katı Olmalıdır");

c++;

}

if (mtxt\_hedef.Text.Length != 3 || Convert.ToInt32(mtxt\_hedef.Text) < 100)

{

MessageBox.Show("Hedef Sayıyı Hatalı Girdiniz");

c++;

}

return c;

}

private void TextEneble(bool a)

{

mtxt\_cift.Enabled = a;

mtxt\_tek1.Enabled = a;

mtxt\_tek2.Enabled = a;

mtxt\_tek3.Enabled = a;

mtxt\_tek4.Enabled = a;

mtxt\_tek5.Enabled = a;

mtxt\_hedef.Enabled = a;

if (a == true)

{

mtxt\_cift.Clear();

mtxt\_tek1.Clear();

mtxt\_tek2.Clear();

mtxt\_tek3.Clear();

mtxt\_tek4.Clear();

mtxt\_tek5.Clear();

mtxt\_hedef.Clear();

}

}

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

{

if (rdb\_random.Checked == true)

{

TextEneble(false);

int[] rteksayi = new int[6];

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

{

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

}

mtxt\_tek1.Text = rteksayi[1].ToString();

mtxt\_tek2.Text = rteksayi[2].ToString();

mtxt\_tek3.Text = rteksayi[3].ToString();

mtxt\_tek4.Text = rteksayi[4].ToString();

mtxt\_tek5.Text = rteksayi[5].ToString();

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

rciftsayi = rciftsayi - rciftsayi % 10;

mtxt\_cift.Text = rciftsayi.ToString();

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

mtxt\_hedef.Text = rhedefsayi.ToString();

}

}

private void rdb\_manuel\_CheckedChanged(object sender, EventArgs e)

{

if (rdb\_manuel.Checked == true)

{

TextEneble(true);

}

}

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

{

Kontrol();

if (c > 0)

{

MessageBox.Show("Hatalı Sayıları Düzeltin");

}

else

{

Hesaplama();

}

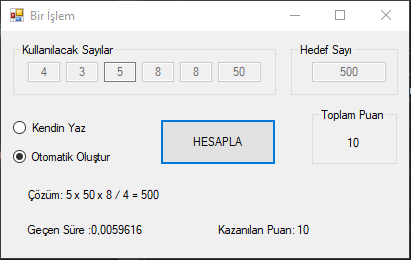
}

}

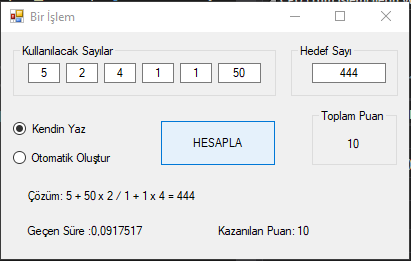
}

**Programın Çalışır Haldeki Görseli**

Otomatik Oluşturma



Kullanıcı Girişiyle Oluşturma



**2) BİR KELİME-FİNAL**

**Çözüm Mantığı**

Kodun temel çalışma prensibi: Kullanıcıdan manuel olarak alınan veya rastgele üretilen harfler bir diziye atanır. Aynı zamanda SQL database imizin içinde harf sayısına göre sıralı bir şekilde bulunan sözlükten kelimeler tek tek alınıp harflere bölünerek yeni bir diziye atanır. Sonra kod içinde elimizde bulunan 2 ayrı dizi bir döngü içinde kıyaslanır. Hedef kelimedeki harfler kendi oluşturduğumuz harflerle birebir eşleşiyorsa kelime ekrana yazdırılır. Eğer bir harf eksik ise onu da joker olarak tanımladığımız string içine atayarak kelimeyi oluşturur ve ekrana yazdırır.

**Kaynak Kodlar**

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;

using System.Data.SqlClient;

namespace Kelime\_Bulma\_Final

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

SqlConnection n = new SqlConnection("Server=DESKTOP-R5P3ANL\\SQLEXPRESS;Database=KelimeBulma;trusted\_connection=true;");

DataSet daset = new DataSet();

public static double secondsTaken = 0;

string kelime;

int j;

int x;

int toplampuan = 0;

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

{

KelimeBulucu();

}

private void KelimeBulucu()

{

string[] harf = new string[9];

DateTime startTime = DateTime.Now;

while (true)

{

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

n.Open();

SqlDataReader dr = comm.ExecuteReader();

while (dr.Read())

{

harf[0] = txt\_bir.Text;

harf[1] = txt\_iki.Text;

harf[2] = txt\_uc.Text;

harf[3] = txt\_dort.Text;

harf[4] = txt\_bes.Text;

harf[5] = txt\_alti.Text;

harf[6] = txt\_yedi.Text;

harf[7] = txt\_sekiz.Text;

j = 0;

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

char[] ayir = kelime.ToCharArray();

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

{

if(Array.IndexOf(harf,ayir[i].ToString())==-1)

{

x = i;

j++;

}

else

{

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

{

if (harf[b] == ayir[i].ToString())

{

harf[b] = " ";

break;

}

}

}

}

if(j==0)

{

harf[8] = ayir[x].ToString();

txt\_bulunankelime.Text = kelime;

PuanHesapla();

break;

}

else if(j==1)

{

harf[8] = ayir[x].ToString();

txt\_jkr.Text = harf[8];

txt\_bulunankelime.Text = kelime;

PuanHesapla();

break;

}

}

dr.Close();

n.Close();

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

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

break;

}

}

private void PuanHesapla()

{

if (txt\_bulunankelime.Text.Length == 9)

{

txt\_kazanılanpuan.Text = 15.ToString();

}

else if (txt\_bulunankelime.Text.Length == 8)

{

txt\_kazanılanpuan.Text = 11.ToString();

}

else if (txt\_bulunankelime.Text.Length == 7)

{

txt\_kazanılanpuan.Text = 9.ToString();

}

else if (txt\_bulunankelime.Text.Length == 6)

{

txt\_kazanılanpuan.Text = 7.ToString();

}

else if (txt\_bulunankelime.Text.Length == 5)

{

txt\_kazanılanpuan.Text = 5.ToString();

}

else if (txt\_bulunankelime.Text.Length == 4)

{

txt\_kazanılanpuan.Text = 4.ToString();

}

else if (txt\_bulunankelime.Text.Length == 3)

{

txt\_kazanılanpuan.Text = 3.ToString();

}

toplampuan += Convert.ToInt32(txt\_kazanılanpuan.Text);

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

}

private void rdb\_rndm\_CheckedChanged(object sender, EventArgs e)

{

Random rastgele = new Random();

string[] uret = new string[8];

string harfler = "abcçdefgğhıijklmnoöprsştuüvyz";

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

{

uret[i] = "";

}

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

{

uret[x] += harfler[rastgele.Next(harfler.Length)];

}

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\_jkr.Text = "?";

}

private void rdb\_kndn\_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\_jkr.Text = "?";

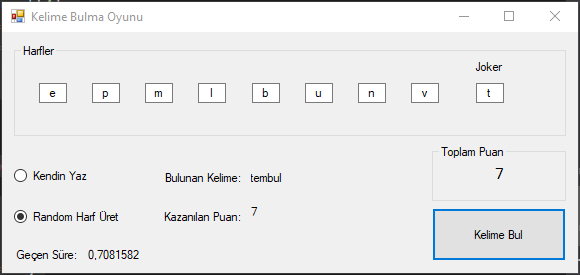
}

}

}

**Programın Çalışır Haldeki Görseli**

Otomatik Oluşturularak



Elle girilerek

