public override string ToString() {

return "(" + this.X + "," + this.Y + ")";

}

// Metoda za izračun površine kvadra

public double Povrsina()

// Statična metoda za izračun volumna kvadra, podanega kot parameter

public static double Volumen(Kvader kvader)

// Bralna lastnost Ploscina, ki vrne površino osnovne ploskve kvadra (dolžina x širina)

public double Ploscina

{

get

{

return dolzina \* sirina;

}

}

int[] izvornoPolje = { 1, 2, 3, 4, 5 };

int[] ciljnoPolje = new int[3];

// Kopiramo prve tri elemente iz izvornega polja v ciljno polje

Array.Copy(izvornoPolje, ciljnoPolje, 3);

// Sedaj ciljnoPolje vsebuje [1, 2, 3]

int[] izvornoPolje = { 1, 2, 3, 4, 5 };

int[] ciljnoPolje = new int[3];

// Kopiramo 2 elementa iz izvornega polja, začenši pri indeksu 2,

// v ciljno polje, začenši pri indeksu 0

Array.Copy(izvornoPolje, 2, ciljnoPolje, 0, 2);

// Sedaj ciljnoPolje vsebuje [3, 4, 0]

## testiranje napak

// Act & Assert

ArgumentException ex = Assert.ThrowsException<ArgumentException>(() => Program.ničleKvEnačbe(a, b, c));

Assert.AreEqual("kompleksne nicle", ex.Message);

}

string.IsNullOrWhiteSpace(posamezne[i

## datoteke:

string vhod = imeVhod;

string izhod = imeIzhod;

using (StreamReader sr = new StreamReader(vhod))

{

using(StreamWriter sw = new StreamWriter(izhod))

{

sw.WriteLine("<table>");

string vrstica;

while((vrstica = sr.ReadLine()) != null)

{

string[] kosi = vrstica.Split(',');

if (kosi.Length == 3)

{

sw.WriteLine("<tr>");

foreach (string kosi2 in kosi)

{

sw.WriteLine("<td>"+kosi2+"</td>");

}

sw.WriteLine("</tr>");

}

}

sw.WriteLine("</table>");

Slika, ki vsebuje besede besedilo, posnetek zaslona, pisava, številka

Opis je samodejno ustvarjen

# Sql

string povNiz = "Server= baza.fmf.uni-lj.si; User Id= student11; Password= student; Database= nobel2012;";

// Povezava s pomočjo Npgsql

NpgsqlConnection pov = new NpgsqlConnection(povNiz);

pov.Open();

// ustrezen ukaz v SQL

string sql = $"SELECT winner FROM nobel where yr={leto} AND subject IN ({subjects})";

NpgsqlCommand ukaz = new NpgsqlCommand();

ukaz.Connection = pov;

ukaz.CommandType = CommandType.Text;

ukaz.CommandText = sql;

NpgsqlDataReader rez = ukaz.ExecuteReader();

while (rez.Read())

{

label2.Text=label2.Text + rez[0].ToString() + "\n";

}

if (label2.Text == "") { label2.Text = "Ni najdena nobena oseba"; };

pov.Close();

=================================================================–––

string povNiz = "Server= baza.fmf.uni-lj.si; User Id= student11; Password= student; Database= nobel2012;";

// Povezava s pomočjo Npgsql

NpgsqlConnection pov = new NpgsqlConnection(povNiz);

pov.Open();

// ustrezen ukaz v SQL

string sql = "SELECT subject FROM nobel group by subject";

NpgsqlCommand ukaz = new NpgsqlCommand();

ukaz.Connection = pov;

ukaz.CommandType = CommandType.Text;

ukaz.CommandText = sql;

NpgsqlDataReader rez = ukaz.ExecuteReader();

string[] subject = new string[8];//ker je 8 področji(vem da ni najboljše tako )

int index = 0;

while (rez.Read())

{

subject[index]=rez[0].ToString();

index++;

}

checkedListBox1.Items.AddRange(subject);

pov.Close();

datoteke:

IDictionary<string, double> slovar = new Dictionary<string, double>();

string pot = @"ig.txt";

StreamReader sr = File.OpenText(pot);

string vrstica =sr.ReadLine();

while (vrstica != null)

{

string[] parts = vrstica.Split(',');

string ime = parts[0];

string[] parts2= vrstica.Split(';');

double vr = double.Parse(parts2[1])\* double.Parse(parts2[2]);

slovar[ime] = vr;

Console.WriteLine(ime+","+vr);

vrstica = sr.ReadLine();

}

sr.Close();

var sortedSlovar = slovar.OrderBy(x => -x.Value);

string pot2 = @"stKoševUrejeno.txt";

StreamWriter dat = File.CreateText(pot2);

foreach(var s in sortedSlovar)

{

dat.WriteLine(""+s.Key+","+ s.Value);

}

dat.Close();